Journal of Universal Computer Science, vol. 21, no. 2 (2015), 268-305
submitted: 11/8/14, accepted: 28/1/15, appeared: 1/2/15 © J.UCS

An Anonymization Algorithm for (α,β,γ,δ)-Social Network Privacy Considering Data Utility

Mehri Rajaei
(Department of Computer Engineering, Iran University of Science and Technology
Tehran, Iran
mrajaei@iust.ac.ir)

Mostafa S. Haghjoo
(Payame Noor University, Kish International Branch, Kish, Iran
haghjoom@iust.ac.ir)

Eynollah Khanjari Miyaneh
(Department of Computer Engineering, Iran University of Science and Technology
Tehran, Iran
khanjari@iust.ac.ir)

Abstract: A well-known privacy-preserving network data publication problem focuses on how to publish social network data while protecting privacy and permitting useful analysis. Designing algorithms that safely transform network data is an active area of research. The process of applying these transformations is called anonymization operation. The authors recently proposed the (α,β,γ,δ)-SNP (Social Network Privacy) model and its an anonymization technique. The present paper introduces a novel anonymization algorithm for the (α,β,γ,δ)-SNP model. The desirability metric between two individuals of social network is defined to show the desirability of locating them in one group keeping in mind privacy and data utility considerations. Next, individuals are grouped using a greedy algorithm based on the values of this metric. This algorithm tries to generate small-sized groups by maximizing the sum of desirability values between members of each group. The proposed algorithm was tested using two real datasets and one synthetic dataset. Experimental results show satisfactory data utility for topological, spectrum and aggregate queries on anonymized data. The results of the proposed algorithm were compared in the topological properties with results of two recently proposed anonymization schemes: Subgraph-wise Perturbation (SP) and Neighborhood Randomization (NR). The results show that the proposed method is better than or similar to SP and NR for preservation of all structural and spectrum properties, except for the clustering coefficient.

Keywords: privacy, network data sharing, anonymization, data utility, information loss, background knowledge.
Categories: H.0, H.2, K.6.5, L.4

1 Introduction

Nowadays, there is a lot of interest by data miners and decision makers to analyze social network data and extract useful knowledge about society [Srivastava 2008], [Kleinberg 2007] such as disease transmission, the influence of a publication, and network data resiliency to faults and attacks. It is difficult to obtain access to actual

network data, in part, because they usually contain private information about individuals (such as relationship with important people) making data owners reluctant to publish them. Instances of the unintended release of private information [Barbaro 2006] have caused organizations to become increasingly conservative about releasing such data sets. The issue becomes how to publish social network data while protecting privacy and permitting useful analysis. This problem is known as *privacy-preserving network data publication*.

There are different types of social networks, including online social network sites, friendship networks, telephone call networks and academic co-authorship networks to name a few. Real-world social network data fits graph data structure. Vertices of the graphs represent individuals and the edges model the relationships between them.

Researches [Bonchi 2011], [Campan 2008], [Cheng 2010], [Cormode 2010], [Hay, 2010a], [Wu 2010b], [Zou 2009] have focused on privacy-preserving network data publication beyond replacing identifiers (such as name and SSN) by a meaningless unique identifier. Malicious users (adversaries) may have background knowledge about some properties of victims (targets) and use them for re-identification to obtain additional information. For instance, they may be able to infer the presence or absence of edges (edge disclosure) or the number of connected entities (degree disclosure).

Preserving the structural properties of graphs is as important as preserving data utility. The present study designed new algorithms to safely transform network data. The process of applying these transformations is called anonymization operation.

Existing anonymization algorithms are based on two models:

1. ***k*-anonymity** [Sweeney 2002]: Some of such algorithms cluster nodes and edges into groups and anonymize a subgraph into a super-node [Bhagat 2009], [Cormode 2010], [Campan 2008], [Hay 2010b]. Each super node contains at least k nodes. In this way they restrict the probability of re-identification to at most $\frac{1}{k}$. Most of these anonymization algorithms are greedy. The nodes are selected in order (based on metrics) to be grouped to make a super node and then continue to generate other groups. Others modify graph structures using a sequence of edge deletions and additions such that each node in the modified graph is indistinguishable with at least k-1 other nodes in terms of some types of structural patterns [Zhou 2011], [Liu 2008], [Wu 2010b], [Zou 2009], [Cheng 2010], [Yuan 2013], [Tai 2014]. In [Wu 2010b], [Yuan 2013], noise nodes may be added to achieve specified requirements. Tai [Tai 2014] splits nodes into multiple substitutes to achieve privacy requirements. All these methods protect re-identification against specified background knowledge. Methods presented in [Zhou 2011], [Yuan 2013] and [Tai 2014] also protect sensitive label and community identity disclosure, respectively, by considering l-diversity [Machanavajjhala 2007] as well as k-anonymity. For l-diversity, each group of nodes should consist of at least l well-represented sensitive values (well-represented means that there are at least l distinct values for a sensitive attribute in each group). Most algorithms in this category use dynamic programming and greedy techniques to apply minimal changes that preserve graph structure as much as possible.

2. ***Randomization:*** Such algorithms modify graph structure by randomly adding/deleting or switching edges. They protect against re-identification in a probabilistic manner [Hay 2007], [Medforth 2011], [Wu 2010a], [Ying 2009], [Ying, 2008], [Bonchi 2011]. Research [Milani Fard 2012, 2013] introduced new edge randomization methods to protect edge disclosure. Only the destination of each edge is replaced with a randomly-chosen node from a subset of nodes (close to the source node of the edge). In this way, the out-degrees of the nodes in published data remain unchanged.

Most researches on network data publishing have been developed to protect against *only one* disclosure. The proposed (α,β,γ,δ)-SNP privacy model [Rajaei 2013] considers both structural and tabular data and protects against disclosure of *membership, sensitive attribute, degree*, and *relationship*. An anonymization technique *ASN* (Ambiguity Social Network) has been proposed based on anatomization operation [Xiao 2006]. *ASN* specifies how to publish data to satisfy (α,β,γ,δ)-SNP privacy requirements. The values of the attributes are published in separate tables.

To make this paper self-contained, the previously proposed privacy model and anonymization technique are briefly reviewed. Next, a greedy anonymization algorithm is proposed that satisfies privacy requirements of (α,β,γ,δ)-SNP and preserves data utility at an acceptable level.

We make following contributions:

- The present paper defines desirability metrics based on privacy and utility requirements to evaluate the desirability of locating two individuals in one anonymization group. This metric considers both tabular and structural properties [section 3].
- It also introduces an anonymization algorithm for ASN technique based on the (α,β,γ,δ)-SNP model. The algorithm uses greedy techniques based on values of the desirability metric between pairs of individuals [section 4].
- The method of estimating query result are presented for three types of queries (aggregate tabular, aggregate network, graph topological and spectrum) on anonymized network data based on ASN [section 6].

Rest of this paper is organized as follows: [section 2] depicts notations and reviews the privacy model and anonymization technique. [Section 3] introduces the proposed desirability metric. [Section 4] describes the proposed anonymization algorithm. [Section 5] describes the three types of queries for evaluating information loss for anonymization algorithm and introduces methods for computing each kind of query for anonymized network data based on ASN. [Section 6] describes the experimental results that demonstrate that information loss from the proposed algorithm is very low.

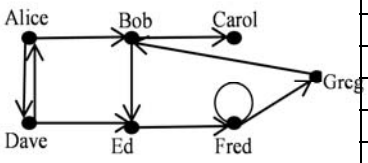
2 Privacy Model and Anonymization Technique

Network data can include tabular and/or structural data. Structural data is used to construct graphs and tabular data comprises the labels of the vertices. The social network data structure is defined formally as follows:

Definition 1: *Social network data* is a simple directed graph $N = (V, E)$ where V is the set of individual attributes ($|V|=n$) and E is the set of relationships on V ($|E|=m$). Elements of E are directed edges or arcs. Multiple nodes cannot represent one individual. Each $v \in V$ has the following labels:

- $I_1, I_2, \ldots, I_r$ are *identifier* attributes such as *name* and *SSN* containing information that explicitly identifies an individual;
- $QI_1, QI_2, \ldots, QI_q$ are *quasi-identifier* attributes (QID) such as *zip code* and *gender* that potentially identify record owners;.
- S describes *sensitive* person-specific information such as *diagnosis* or *income* that is assumed to be unknown to adversaries.

[Fig. 1] is an example of a money transformation network. Each node contains five labels from one of three categories: *name* (identifier), *gender*, *job* and *zip code* (quasi identifier) and *income* (sensitive).



name	gender	Job	zipcode	Income
Alice	F	Nurse	23000	3500
Bob	M	Teacher	21000	3000
Carol	F	Nurse	52000	4500
Dave	M	Teacher	73000	4000
Ed	F	Doctor	23000	7000
Fred	M	Mechanic	73000	5000
Greg	M	Police	43000	7000

Figure 1: An example of money transformation network

An adversary may know the quasi-identifier labels, in-degrees out-degrees, and sensitive values about victims. The goal of the proposed privacy model is to restrict the likelihood of extracting new information about victims under specified thresholds. It restricts the probability of assigning a specific sensitive value, in-degree value, out-degree value, and existence of a directed relationship to a specified individual when adversary has some of above background knowledge. It also protects against membership disclosure, which decreases the certainty of other findings. The (α,β)-privacy [Wang 2010] and *l*-diversity [Machanavajjhala 2007] models are extended for network data in the proposed privacy model.

Definition 2 (α,β,γ,δ)*-SNP*: Given a network data N, let N^* be its anonymized version. Say that N^* satisfies (α,β,γ,*δ)-SNP* (Social Network Privacy) if it satisfies all of the following constraints:

- α-presence: for each entity $i \in N^*$ ($i[QI] = \{qi_1, qi_2, \ldots, qi_q\}$), $\Pr(i \in N) \leq \alpha$ ("Pr" denotes probability);
- β-sensitive-association: for any sensitive association $(i, s) \in N^*$, $\Pr((i, s) \in N | i \in N) \leq \beta$;
- γ-degree-association: for any in-degree (out-degree) association $(i, d) \in N^*$, $\Pr((i, d) \in N | i \in N) \leq \gamma$;

- δ-relationship: for any directed relationship $(i_1, i_2) \in N^*$, $\Pr((i_1, i_2) \in N | i_1 \in N, i_2 \in N) \leq \delta$.

$\Pr(i \in N)$ denotes the probability of adversary knowledge about the existence of individual i (with specified quasi-identifier attributes) in the original network data. The α-presence limits $\Pr(i \in N)$ to below α.

β-sensitive-association restricts adversary belief about an association between individuals and the value of sensitive attribute S to below β. The association between individual i and sensitive value s is denoted by (i, s). Since inference of any private association of a specific individual is based on the presence of his/her nodes in the original network, probability of sensitive-association privacy is defined as conditionally dependent on the probability of presence privacy. The γ-degree-association is defined as similar to the β-sensitive-association for in-degree (out-degree). The δ-relationship binds adversary knowledge of the existence of a directed edge from one individual to another in the network to δ. (i_1, i_2) denotes a directed relationship from individual i_1 to i_2. Similarly, the probability of relationship disclosure is conditionally dependent on the probability of the presence of both individuals.

The proposed anonymization technique specifies how to publish data to satisfy (α,β,γ,δ)*-SNP* privacy requirements. This technique stores all tabular and structural data in multiple relational tables. Instead of publishing a generalized value [Sweeney 2002] for each attribute, exact values are published in separate tables. In this way, the number of individuals generated by the lossy join of these tables is more than the number of individuals in the original data. In other words, false tuples are generated by lossy join creating uncertainty about membership and assigning of private information to each person. A formal definition of *ASN* follows.

Definition 3 (*ASN(Anonymity Social Network)*): Given social network data $N = (V, E)$, assume that all identifier attributes $I_1, I_2, \ldots, I_r$ are removed and one unique and random label l is assigned to each vertex v. Vertices are partitioned into n' groups $G = \{G_1, G_2, \ldots, G_{n'}\}$ such that $\bigcup_{i=1}^{n'} G_i = V$, and for any $i \neq j, G_i \cap G_j = \emptyset$. *ASN* produces two quasi-identifier auxiliary tables (QATs), a sensitive table (ST), a degree table (DT) and a successor vertices table (SVT) as:

- Quasi-identifier attribute set $Q = \{QI_1. QI_2, \ldots, QI_q\}$ is partitioned into two sets $P = \{P_1, P_2\}$ such that $\forall i, j: P_i \cap P_j = \emptyset$ and $\bigcup_{i=1}^{2} P_i = Q$. Each set $P_i = \{QI_{i,1}, QI_{i,2}, \ldots, QI_{i,|Pi|}\}$ ($|P_i|$ denotes the size of P_i) corresponds to auxiliary table QAT_i of schema $(GID, QI_{i,1}, \ldots, QI_{i,|Pi|}, \text{count})$ with $|P_i| + 2$ columns. For any group $G_j (1 \leq j \leq n')$ and any distinct quasi-identifier value $(q_1, q_2, \ldots, q_{|Pi|})$ of $(QI_{i,1}, QI_{i,2}, \ldots, QI_{i,|Pi|})$ in G_j, there is a tuple $(j, q_1, q_2, \ldots, q_{|P_i|}, c) \in QAT_i$ where c is the number of vertices $v \in G_j$ such that $v[QI_{i,1}] = q_1, v[QI_{i,2}] = q_2, \ldots, v[QI_{i,|P_i|}] = q_{|P_i|}$ ($v[A]$ is the value of attribute A of vertex v).
- S corresponds to the ST of schema (GID, S, count). For any group $G_j (1 \leq j \leq n')$ and any distinct sensitive value s of S in G_j, there is a tuple

$(j, s, c) \in ST$, where c is the number of vertices $v \in G_j$ such that $v[\mathrm{S}] = s$.

- The exact in-degree and out-degree of the vertices is published in the DT of schema $(GID, label, Din, Dout)$. For any group $G_j (1 \leq j \leq n')$ and any vertex v in G_j, there is a tuple $(j, l, din, dout) \in DT$, where din and dout are the in-degree and out-degree of v such that $v[label] = l$.
- Successor vertices of nodes of each group are published in a SVT of schema $(GID, label, count)$. For any group $G_j (1 \leq j \leq n')$ and any distinct vertex u in the successor nodes of G_j, there is a tuple $(j, l, c) \in SVT$, where c is the number of vertices $v \in G_j$ such that $u \in v[succ]$ and $u[label] = l$, where $v[succ] = \{u | (v, u) \in E\}$.

[Fig. 2] shows an example of the *ASN* technique using the network data in [Fig. 1].

When an adversary tries to discover new information about victim *i*, he should find all groups that cover *i* based on his background knowledge about *i*. If background knowledge of the adversary about the victim is $\mathrm{X}_1 = \mathrm{v}_1, \ldots, \mathrm{X}_b = \mathrm{v}_b$ where $\mathrm{X}_1, \ldots, \mathrm{X}_b$ are known properties, then group ID of all covering groups equals $\pi_{Gid}\left(\sigma_{\mathrm{X}_1=\mathrm{v}_1 \wedge \ldots \wedge \mathrm{X}_b=\mathrm{v}_b}(QAT_1 \bowtie QAT_2 \bowtie DT \bowtie ST)\right)$. These two reasons means that victim *i* only belongs to one group. Based on Definition 1, there are no multiple nodes related to one individual in the network data. Based on Definition 7 below, the intersection of the groups in *ASN* is empty.

Gid	gender	job	count
1	F	nurse	2
1	F	doctor	1
1	M	mechanic	1
2	M	teacher	2
2	M	police	1

QAT_1

Gid	zipcode	Count
1	23000	2
1	52000	1
1	73000	1
2	21000	1
2	73000	1
2	43000	1

QAT_2

Gid	income	count
1	3500	1
1	4500	1
1	5000	1
1	7000	1
2	3000	1
2	4000	1
2	7000	1

ST

Gid	label	Din	Dout
1	a	1	2
1	c	1	0
1	e	2	1
1	g	1	1
2	b	2	2
2	d	1	2
2	f	2	2

DT

Gid	label	Count
1	b	2
1	f	1
1	d	1
2	c	1
2	g	1
2	e	2
2	f	1
2	a	1

SVT

Figure 2: An example of ASN technique

When an adversary tries to reconstruct quasi-identifier values, he/she will have multiple candidates resulting from the lossy join of QATs on the GID. The set of all candidates for each group G_j, is called *Generated Combinations* set (GC_j), where $|GC_j| = \pi_{count(*)}(\sigma_{GID=j} QAT_1) \times \pi_{count(*)}(\sigma_{GID=j} QAT_2)$. Based on the sum of *count* values for each group *j*, $|G_j| = \pi_{sum(count)}(\sigma_{GID=j} QAT_1)$, so there are $\binom{|GC_j|}{|G_j|}$ choices for reconstructing members of G_j.

In some of these choices, the number of repetitions of each distinct value matches the *count* value of that in the QATs. These sets of choices are called *probable World* G_j (PW_j). Of all possible worlds, only one subset contains the same individuals as in

the original data. Some of these possible worlds contain quasi-identifiers of individual i and is called *interesting worlds* ($IW_j^i = \{T | T \in PW_j,$ and $i \in T\}$, so $\Pr(i \in G_j) = \frac{|IW_j^i|}{|PW_j|}$. Each member of GC_j may belong to group G_j, but the maximum size of IW_j^i in G_j is a combination of the most frequent value of each QAT in G_j (*probable combination* (pc_j)). The most frequent value of QAT_i in G_j is $\pi_{QID_{i,1},\ldots,QID_{i,|Pi|}}\left(\sigma_{Gid=j \wedge count=\pi_{\max(count)}(\sigma_{Gid=j}(QAT_i))}(QAT_i)\right)$. In other words, $\forall i \in GC_j$: $|IW_j^i| \leq |IW_j^{pc_j}|$, so $\forall i \in GC_i$: $\Pr(i \in G_j) \leq \frac{|IW_j^{pc_j}|}{|PW_j|}$.

Let $G^* = \{G_1, \ldots, G_k\}$ be the set of all groups that cover individual i based on quasi-identifier values. Since i belongs to at most one group, the maximum probability of presence for i is $\Pr(i \in N) \leq \max_{\forall G_j \in G^*} \Pr(i \in G_j) \leq \max_{\forall G_j \in G^*} \frac{|IW_j^{pc_j}|}{|PW_j|}$.

If an adversary knows that i belongs to G_j, then the probability of associating i with sensitive value s is $\frac{C_j^s}{|G_j|}$ where $C_j^s = \pi_{count}(\sigma_{GID=j \wedge S=s} ST)$. When sensitive value s is the most frequent sensitive value (f_j) in group G_j, maximum sensitive-association probability occurs for (i, f_j) in that group. Since $C_j^{f_j} = \pi_{max(count)}(\sigma_{GID=j} ST)$, $\Pr\left((i,s) \in G_j \middle| i \in G_j\right) \leq \frac{C_j^{f_j}}{|G_j|}$. Let $G^* = \{G_1, \ldots, G_k\}$ be the set of all groups that cover individual i. Then $\Pr((i,s) \in N | i \in N) \leq max_{\forall G_j \in G^*}\left\{\Pr\left((i,s) \in G_j \middle| i \in G_j\right)\right\} \leq max_{\forall G_j \in G^*}\left\{\frac{C_j^{f_j}}{|G_j|}\right\}$.

Same as sensitive-association probability, the maximum probability of assigning degree d to an in-degree (out-degree) of individual i is $\Pr((i,d) \in N | i \in N) \leq max_{\forall G_j \in G^*}\{\frac{Cin_j^{fin_j}}{|G_j|}\}$ ($\Pr((i,d) \in N | i \in N) \leq max_{\forall G_j \in G^*}\{\frac{Cout_j^{fout_j}}{|G_j|}\}$) where $Cin_j^{fin_j}$ is the frequency of the most frequent in-degree (fin_j) in group G_j and $Cout_j^{fout_j}$ is the frequency of the most frequent out-degree($fout_j$) in G_j.

When an adversary tries to reconstruct output edges from each group based on the DT and SVT tables, he/she will have multiple candidates from the lossy join of the tables on GID (as for presence probability). The set of all edges generated by lossy join for G_j is GE_j, where $|GE_j| = \pi_{count(*)}(\sigma_{GID=j} DT) \times \pi_{count(*)}(\sigma_{GID=j} SVT)$. The sum of *Dout* for each group j makes the total number of output edges (OE) from group j, $|OE_j| = \pi_{sum(Dout)}(\sigma_{GID=j} DT)$. There are $\binom{|GE_j|}{|OE_j|}$ choices to reconstruct the output edges of G_j. Some of these choices are valid; each valid choice $VC = \{(u,w) | (u,w) \in GE_j\}$ ($|VC|=|OE_j|$) should meet two requirements. 1)Let $S(u, VC) = \{(u,w) | (u,w) \in VC\}$ be the set of all directed output edges of VC with u being the

source node. Now, for each label u in G_j, $|S(u,VC)|$should equal the *D*out for u in DT. 2) Let $d(u,VC) = \{(w,u)|(w,u) \in VC\}$ be the set of all directed output edges of VC with u being the destination node. For each label u in G_j, $|d(u,VC)|$should equal the *count* value in the SVT table for G_j. These sets of choices are called $possible\ world\ edges\ of\ G_j\ (PWE_j)$. Some of these possible worlds contain (i_1,i_2), called interesting world edges ($IWE_j^{(i_1,i_2)} = \{T|T \in PWE_j, \text{and } (i_1,i_2) \in T\}$; thus, $\Pr((i_1,i_2) \in |\mathrm{OE}_j||i_1 \in \mathrm{G}_j, i_2 \in N) = \frac{|IWE_j^{(i_1,i_2)}|}{|PWE_j|}$. The number of interesting worlds contain most probable edge (*pe*) (the edge from the label with maximum *Dout* in DT to the most frequent label as the successor of nodes of G_j in SVT) is greater than for other combinations of GE_j. $\forall(i_1,i_2) \in \mathrm{GE}_j: \Pr((i_1,i_2) \in |\mathrm{OE}_j||i_1 \in \mathrm{G}_j, i_2 \in N) \leq \frac{|IWE_j^{pe_j}|}{|PWE_j|}$. If i_2 does not belong to the successor nodes of G_j, this probability is zero.$\Pr((i_1,i_2) \in N|i_1 \in N, i_2 \in N) \leq max_{\forall G_j \in G^*}\{Pr((i_1,i_2) \in |\mathrm{OE}_j||i_1 \in \mathrm{G}_j, i_2 \in N)\} \leq max_{\forall G_j \in G^*}\{\frac{|IWE_j^{pe_j}|}{|PWE_j|}\}$.

Corollary. Let N be network data and let N^* be its anonymized network; using *ASN* technique with tables N^* = {QAT_1, QAT_2, ST, DT, SVT} and n' groups G = {G_1, G_2,…,$G_{n'}$}. N^* satisfies (α,β,γ,δ)*-SNP* privacy requirements if $\forall G_j \in G: \frac{|IW_j^{pc_j}|}{|PW_j|} \leq \alpha, \frac{C_j^{f_j}}{|G_j|} \leq \beta, \frac{Cin_j^{fin_j}}{|G_j|} \leq \gamma, \frac{Cout_j^{fout_j}}{|G_j|} \leq \gamma, \frac{|IWE_j^{pe_j}|}{|PWE_j|} \leq \delta$.

3 Desirability Metric

The *Desirability Metric* (DM) measures the desirability of locating two individuals in one group with the goal of attaining privacy requirements as soon as possible. We assign a desirability weight between each two individuals based on this metric. We should consider two aspects of grouping of individuals: the privacy of members and data utility. So we define two metrics *Privacy Metric* (*PM*), and *Utility Metric* (*UM*). PM is determined based on privacy requirements; this metric shows how different are the properties of two individuals. UM is computed based on topological and aggregate properties and show how similar are the properties of successor nodes of two individuals. Therefore, DM equals the sum of the privacy metric (PM), and utility metric (UM). This produces:

$$DM(v,u) = PM(v,u) + UM(v,u) \quad (1)$$

3.1 Privacy metric

PM covers the following four cases:

1. As shown in section 2, the maximum possibility of presence in G_j is $\frac{\left|IW_j^{pc_j}\right|}{|PW_j|}$. Let for *i=1,2* $X_i = \pi_{GID=j}(QAT_i)$. For a constant $|G_j|$, if the number of distinct values of each QID-attribute ($|X_i|$) in each group increases, the maximum possibility of presence will decrease for two reasons. First, since for *i=1,2* $\left|G_j\right| = \sum_{x \in X_i} x[count]$(the sum of $x[count]$ remains unchanged while $|X_i|$ increases), the value of $x[count]$ decreases for $x \in X_i$. If the frequency of the most frequent value ($\max_{x \in X_i}\{x[count]\}$) decreases, the difference between the frequency of the most frequent value and that of the least frequent value decreases and maximum possibility of presence decreases. Second, when the *count* values of all distinct values are 1, the number of valid choices of G_j is maximized. By decreasing *count* values, the denominator increases. For example, locate *Alice*, *Bob*, and *Carol* in a group [Fig. 1]. The tuple values of QAT_1 and QAT_2 for this group are {(F,nurse,2),(M,Teacher,1)} and {(23000,1),(21000,1),(52000,1)} respectively. Its possible world is:
$$\left\{\begin{array}{l}\big((\text{F},\text{nurse},23000),(\text{F},\text{nurse},21000),(\text{M},\text{Teacher},52000)\big),\\ \big((\text{F},\text{nurse},23000),(\text{F},\text{nurse},52000),(\text{M},\text{Teacher},21000)\big),\\ \big((\text{F},\text{nurse},21000),(\text{F},\text{nurse},52000),(\text{M},\text{Teacher},23000)\big)\end{array}\right\}$$
$(\text{F},\text{nurse},23000)$ is one probable combination that appears in 2 cases. The maximum possibility of presence in this group is $\frac{2}{3}$. Now locate *Fred* in this group instead of *carol*. There are more distinct values in QAT_1 with the same group size. Here, the size of the possible world increases to 6:
$$\left\{\begin{array}{l}\big((\text{F},\text{nurse},23000),(\text{M},\text{Teacher},21000),(\text{M},\text{Mechanic},73000)\big),\\ \big((\text{F},\text{nurse},23000),(\text{M},\text{Teacher},73000),(\text{M},\text{Mechanic},21000)\big),\\ \big((\text{F},\text{nurse},21000),(\text{M},\text{Teacher},73000),(\text{M},\text{Mechanic},23000)\big),\\ \big((\text{F},\text{nurse},21000),(\text{M},\text{Teacher},23000),(\text{M},\text{Mechanic},73000)\big),\\ \big((\text{F},\text{nurse},73000),(\text{M},\text{Teacher},23000),(\text{M},\text{Mechanic},21000)\big),\\ \big((\text{F},\text{nurse},73000),(\text{M},\text{Teacher},21000),(\text{M},\text{Mechanic},23000)\big)\end{array}\right\}$$
The appearance of all combinations equals 2. In other words, the difference between the frequency of the most frequent value and that of the least frequent value is zero. The possibility of the presence of each combination in this group is $\frac{1}{3}$; therefore, $PM(v,u)$ increases for each different QID-value for *v* and *u*.
2. As shown, the maximum possibility of sensitive association for G_j is $\frac{C_j^{f_j}}{|G_j|}$. If $C_j^{f_j}$decreases, the possibility of association also decreases. In other words, increasing the number of distinct values for the sensitive attribute of individuals of each group of constant size decreases the possibility of sensitive association in that QID group. $PM(v,u)$ increases when the sensitive values of *v* and *u* are different.
3. As shown, the maximum possibility of degree-association in G_j is $\max\left\{\frac{Cin_j^{fin_j}}{|G_j|}, \frac{Cout_j^{fout_j}}{|G_j|}\right\}$. Again, increasing the number of distinct values for the in-

degrees (out-degrees) of individuals of each group of fixed size decreases the probability of degree-association. Since in-degree and out-degree are numerical attributes, in-degree values for individuals of one group can occur in a narrow range (for example, $[10, 15]$); therefore, the adversary may infer a narrow range for the in-degree but not an exact value; thus, wide ranges for in-degree (out-degree) values for members of each group are preferred.

4. As proven, the maximum possibility of edge disclosure in G_j is $\frac{\left|IWE_j^{pe_j}\right|}{|PWE_j|}$. The method of measuring the probability of a relationship is similar to that for the possibility of presence; however, to decrease *count* value of QAT, the number of distinct values of each QID attribute in each group of constant size should increase. The role of *count* in QATs is the same as *Dout* in DT and *count* in SVT. In this case, the same policy is not applicable because the out-degrees of nodes cannot be changed nor can only nodes with low out-degrees be used. To decrease the maximum possibility of edge disclosure, it is best to decrease the difference between probabilities of disclosure of the most probable edge with disclosure of the least probable edge. There are two options for this. First, decrease the difference between the out-degrees of the nodes of each group j with their maximum out-degrees. For this goal, the out-degrees of the nodes of each group should fall into a narrow range. This goal is in conflict with the tendency for degree-association. For example, let a two-member group have labels *a* and *b* with out-degrees 1 and 3, respectively. This group has four distinct successors (s, t, q, r) having frequency 1. The size of the possible world edges for this group is 4. Edge (*b,s*) is one probable edge that appears in 3 cases. The maximum probability of relationship disclosure in this group is $\frac{3}{4}$. Now, suppose that the out-degrees of both of *a* and *b* are 2 (with a lower difference between out-degrees). In this case, |PWE| = 6 and $|IW^{pe}|$ = 3. The maximum probability of relationship disclosure decreases to $\frac{1}{2}$.The second option is to decrease the *count* value of all successor labels of G_j in SVT. Grouping nodes with lower common successor nodes is advantageous so that the maximum possibility of edge disclosure decreases.

According to the above reasoning, grouping individuals with greater differences between attributes, in-degrees, and successor nodes is more desirable. In this way, groups with smaller group sizes attain α-presence, β-sensitive-association, γ-degree-association, and δ-relationship; thus, PMs of individual pairs with a greater number of different properties are assigned higher values.

Definition 4 (*Privacy Metric(PM)*): Let $N = (V, E)$ be original social network data. For each two individuals *u, v* in *V*, $PM(v, u)$ can be calculated as:

$$PM(v,u) = \sum_{k=1}^{q} \left(F_{QI} \times Pe_{QI_k} \times not\_equal(v[QI_k], u[QI_k])\right) + \left(F_S \times Pe_s \times not\_equal(v[S], u[S])\right) + \left(F_{Din} \times Pe_{Din} \times \frac{|v[Din] - u[Din]|}{max(Din) - \min(Din)}\right) + \left(\begin{array}{c} F_{Dout} \times Pe_{Dout} \times \frac{|v[Dout] - u[Dout]|}{max(Dout) - \min(Dout)} + \\ F_{succ} \times (1 - Pe_{Dout}) \times \left(1 - \frac{|v[Dout] - u[Dout]|}{max(v[Dout], u[Dout])}\right) \end{array}\right) + \left(F_{succ} \times Pe_{succ} \times \left(1 - \frac{max(v[Dout], u[Dout])}{|v[succ] \cup u[succ]|}\right)\right) \quad (2)$$

Where

$$not\_equal(x,y) = \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$$

$$\forall x \in \{QI_1, \dots, QI_q, S, Dout, Din\}: Pe_x = \frac{\sum_{\forall v \in d_x} \binom{frq_v^x}{2}}{\binom{n}{2}}$$

d_x is the set of all distinct values of property x that appears more than once in network data N

$$frq_v^x = \pi_{count(*)}(\sigma_{x=v}(N))$$

$$Pe_{succ} = \frac{\sum_{\forall v \in V} \sum_{\forall u \neq v \in V} has\_intersect(v[succ], u[succ])}{2 \times \binom{n}{2}}$$

$$has\_intersect(X,Y) = \begin{cases} 0 & X \cap Y = \emptyset \\ 1 & X \cap Y \neq \emptyset \end{cases}$$

$F_{QI}, F_S, F_{Din}, F_{Dout}, F_{succ}$ are specified by data publisher.

Since the difference in properties with limited domains and less distinct values is more important than the difference in properties with wide domains and more distinct values, use $Pe_x$where $x \in \{QI_1, \dots, QI_q, S, Dout, Din\}$ to determine the probability of equality of property *x* for two individuals. A higher Pe_x indicates the importance of differences in property *x* for individuals of each group. For example, the *gender* attribute can only have *F* or *M* values. The probability of the same *gender* values for two individuals is high. On the other hand, *zip code* can get a wider range of values, so the probability of the same *zip code* value for two individuals is low.

To calculate Pe_x, first find d_x (set of all distinct values of property *x* that appear more than once in network data *N*). For each value $v \in d_x$, the probability of the values of property *x* for two individuals from *N* having the same value *v* is $\frac{\binom{frq_v^x}{2}}{\binom{n}{2}}$ where frq_v^x is the number of repetitions of value *v* in property *x* in network data *N*. The probability of two individuals having the same value for *x* is $\sum_{\forall v \in d_x} \frac{\binom{frq_v^x}{2}}{\binom{n}{2}}$. The difference in the value of property *x* having higher Pe_x has more effect on PM. For

example, in the dataset shown in [Fig. 1], there are 4 and 3 tuples with values *M* and F respectively, for the *gender* attribute; thus $Pe_{gender} = \frac{\binom{4}{2}+\binom{3}{2}}{\binom{7}{2}} = \frac{9}{21}$. There are two values for *zip code* that appear twice $(23000, 73000)$; thus, $Pe_{zipcode} = \frac{\binom{2}{2}+\binom{2}{2}}{\binom{7}{2}} = \frac{2}{21}$.

The function not-equal checks the equality of its arguments. It can be modified for numerically-sensitive values such as salary to protect against range attacks [Zhang 2007]. Sometimes numerical sensitive attributes for each group may contain *k* distinct values, but all these values occur in a narrow range, which helps attackers infer the range without specifying the exact value. For example, there are 5 individuals with different salaries $\{1000, 1050, 1070, 1100, 1020\}$ in one group. Although their salaries are different, all fall in a narrow range $[1000,1100]$. To protect against this attack, the values for the sensitive attribute of each group should cover a wide range.

When computing PM, modify the not-equal function for numerically sensitive values to $not\_equal(v[S], u[S]) = \frac{|v[S]-u[S]|}{\max(S)-\min(S)}$, where max(S) and min(S) are the maximum and minimum values, respectively, for sensitive attributes in network data *N*. This fraction measures the difference in their values compare to the available range in the dataset as a number between 0 to 1.The resulting fraction for small differences in the values of *v* and *u* for a narrow-range attribute has a greater effect than that for a wide-range attribute.

The in-degree of an individual is also numerical and should be protected against an adversary. Its *not-equal* function is $\frac{|v[Din]-u[Din]|}{max(Din)-\min(Din)}$; thus a greater difference in in-degrees relative to the range of all in-degree nodes increases the value for the *not-equal* function and consequently has more effect on PM.

As stated, out-degrees of group members affect the requirements of degree-association and relationship. There are two conflicting policies: 1) The out-degree values of members of one group should be in a wide range to protect against range attack. To prevent out-degree disclosure, it is sufficient to have different values to satisfy δ-degree-association. 2) On the other hand, out-degrees of group members should be close together to protect relationship disclosure. To resolve the conflict, the effect of out-degree on PM for any two individuals should cover both cases. Its first effect is similar to in-degree at $F_{Dout} \times Pe_{Dout} \times \frac{|v[Dout]-u[Dout]|}{max(Dout)-\min(Dout)}$. Its second effect is calculated as $F_{succ} \times (1 - Pe_{Dout}) \times \left(1 - \frac{|v[Dout]-u[Dout]|}{max(v[Dout],u[Dout])}\right)$. In this way, the coefficients of F_{Dout} and F_{succ} specify their effects.

When the out-degrees of group members are low, a small difference in out-degree results in a large change in relationship disclosure probability for that group. For example, in a group with 3 members and out-degree sequence (2,2,1) in a DT with 5 different individuals as successor nodes in SVT with no intersection in their successor nodes, there are 30 valid choices to reconstruct output edges; 12 of them contain probable combination. The maximum probability of disclosure of each possible output edge is $\frac{12}{30} = \frac{2}{5}$. If this group contains individuals with out-degree sequence (3,1,1) in DT with 5 different individuals as successor nodes in SVT, there are 20 valid choices to reconstruct output edges and 12 of them contain a probable combination. The maximum probability of disclosure of each possible output edge is

$\frac{12}{20} = \frac{3}{5}$. As shown in this example, small changes in the out-degree of one group can change the disclosure probability by about $\frac{1}{5}$. If the out-degree sequence is (9,10,10), and there is no intersection between successor nodes of individuals of this group, the maximum relationship disclosure likelihood is $\frac{10}{29}$ (the SVT table contains 29 labels with frequency 1 for this group). If the out-degree sequence is (9,9,11) with 29 distinct successors, the maximum likelihood of relationship disclosure changes to $\frac{11}{29}$. In this case, a small change in out-degree causes a small change of about $\frac{1}{29}$ in the probability of relationship disclosure.

For this reason, divide the difference in out-degrees of two individuals by their maximum out-degree. For high out-degree nodes, a small difference in out-degree does not drastically decrease PM. The fraction $\frac{|v[Dout]-u[Dout]|}{max(v[Dout],u[Dout])}$ falls in the range [0,1]. When out-degrees of both individuals are zero, consider this fraction to be zero.

As stated, to protect against relationship disclosure, the out-degrees of nodes of each group should fall into a narrow range. It is better to keep this fraction near 0; therefore, use $\left(1 - \frac{|v[Dout]-u[Dout]|}{max(v[Dout],u[Dout])}\right)$. The higher value for this expression shows more desirability for putting the two individuals into one group and increases their PM value. Since a higher probability of inequality of out-degree between individuals $(1 - Pe_{Dout})$ makes it harder to put individuals with the same out-degree in one group, that expression is multiplied by$(1 - Pe_{Dout})$.

Pe_{succ} indicates the probability of having the same successor for two individuals in network data *N*. To compute it, count all pairs of individuals *N* with common successor nodes and divide them by the number of all possible pairs. A higher value for Pe_{succ} means a low probability of finding individuals without a common successor. If individuals *v* and *u* are located in a group, there are $|v[succ] \cup u[succ]|$ distinct labels in SVT for that group. The high bound for probability of existence of a directed relationship from *v* to each member of $v[succ] \cup u[succ]$ is approximately $\frac{v[Dout]}{|v[succ] \cup u[succ]|}$. For example, let there be two nodes having out-degrees 2 and 3 where the union of their successor nodes has 4 members. If these two nodes are located in one group, since the out-degree of the second node is 3, it should be connected to 3 successors; the probability of existence of an edge between it and each member of the successors is $\frac{3}{4}$.

A simple expression was used to compute PM. The fraction $\frac{max(v[Dout],u[Dout])}{|v[succ] \cup u[succ]|}$ approximates the maximum probability of disclosure of a directed relationship if these two individuals are members of one group. Lower values denote increased desirability to put them in one group. $F_{succ} \times Pe_{succ} \times \left(1 - \frac{max(v[Dout],u[Dout])}{|v[succ] \cup u[succ]|}\right)$ is used in the PM equation. As for the previous case, if the out-degree of two individuals equals zero, consider $\frac{max(v[Dout],u[Dout])}{|v[succ] \cup u[succ]|}$ to be zero. Since there are no successor nodes, the probability of disclosure of the relationship is zero.

$F_{QI}, F_S, F_{Din}, F_{Dout}$ and $F_{succ}$are coefficients for quasi-identifiers, sensitive attribute, in-degree, out-degree, and relationship, respectively, which have important

effects on PM and grouping methods. The diversity of sensitive attributes, in-degree (out-degree), and successor individuals has a direct effect on sensitive-association, degree-association, and relationship probabilities, respectively. Their coefficients should be greater than F_{QI}, which effects PM for all quasi-identifier attributes; thus, if the number of quasi-identifier attributes is high, the increased value of PM relates to inequality in the quasi-identifier attributes. The composition of quasi-identifiers only affects probability of presence, while other properties have a direct effect on other privacy constraints. As a result, if other coefficients are set to ~1, F_{QI} should be set to about $\frac{1}{q}$, where q is the number of quasi-identifiers.

When the corresponding quasi-identifier or sensitive attribute of two individuals are not equal, the output of the *not-equal* function is 1. For other properties, when the corresponding values are not equal, the inequality ratio is in the range [0,1]. To compensate, F_{Din}, F_{Dout} and F_{succ} coefficients should be greater than 1.

Other important factors used to set the coefficients are thresholds α, β, γ and δ. For example, when the constraint of the β-sensitive-association is limited and the density of the sensitive attribute is high and close to β, it is better to choose a higher value for F_S. This increases the effect of differences in sensitive values of individuals. The density of x is the ratio of the number of individuals with the most frequent value for property x by the number of individuals in the network data. For instance, if the density of the successor property attribute is 20%, it means that 20% of individuals have the same successor node. Thresholds β, γ and δ cannot be less than the density of the sensitive attribute, in-degree (out-degree), and successor properties, respectively. It their value is less than their density, then no grouping will be found thatall groups satisfy the privacy requirements. It is reasonable to consider high coefficients for more limited constraints.

As stated in Definition 4, the calculation for PM is $O(q)$ (q is the number of QID attributes of N) and there are $O(n^2)$ pairs of individuals; the total time complexity for computing PM for all pairs is $O(qn^2)$. In addition, computing each factor Pe_x where $x \in \{QI_1, \dots, QI_q, S, Dout, Din\}$ is $O(n)$ and the time complexity for computing Pe_{succ} is $O(n^2 * max(outdegree) * log(max(outdegree)))$. The upper bound for the *has-intersect* time complexity is $O(d * logd)$ where d is the maximum size of its input parameters. In the worst case, d equals *max(out-degree)*. Pe_x and Pe_{succ} are calculated only once.

3.2 Utility metric

This metric determines the similarity of two individuals in the data and structural utility. [Section 5] reviews four types of analysis possible for published data. When publishing data, the results of analysis on published data should be similar to those on original data. For PM, the value of DM is increased to satisfy the privacy requirements of groups of minimum size because a smaller size for the anonymization group decreases information loss. In UM, the value of DM is increased to preserve better data utility, while considering two concerns:

1. Preserving the shortest path length (distance) between each two individuals in the published network is an important metric in topological and structural properties of the network data. When an analyzer reconstructs graph data

from published ASN data, each node label in the SVT table for group j can be considered a successor node for each member of group j. The goal is to group members with low average distances from all successor nodes of that group. If the shortest path length is preserved, closeness can also be preserved, along with the diameter and spectrum properties of the graph data.

2. An aggregate network query calculates aggregation on paths or subgraphs satisfying specific query conditions. One example is the average distance from nodes with specified values for sensitive attributes to other nodes in a network structure. To preserve the data utility of this type of query, the members are grouped with the most similar sensitive value distributions of their successor nodes.

Based on above reasoning, higher value is assigned to utility metric of the two individuals with lower average distances to their successor nodes and more similar sensitive value distributions of successor nodes.

Definition 5. (Utility *Metric*(*UM*)): Let $N = (V, E)$ be the original social network data, for each two individuals u and v in V, $UM\ (v,u)$ is calculated as:

$$UM(u,v) = F_{top}\frac{|OU| + |OV|}{\sum_{x\in OU} dist(v,x) + \sum_{x\in OV} dist(u,x)} + F_{an}\left(1 - \frac{\sum_{i\in SValSet}\frac{|p_v(i) - p_u(i)|}{max(p_v(i), p_u(i))}}{|SValSet|}\right) \quad (3)$$

where

$OV = v[succ] - u[succ]$

$OU = u[succ] - v[succ]$

$dist(v,u) = \begin{cases} 2n, & \text{there is not path from } v \text{ to } u \\ \text{length of shortest path from } v \text{ to } u \text{ on N}, & otherwise \end{cases}$

$SValSet = \{s | \exists x \in OV \cup OU : x[S] = s\}$

$\forall i \in SValSet: p_v(i) = \frac{|\{x|x\in OV: x[S]=i\}|}{|OV|}, p_u(i) = \frac{|\{x|x\in OU: x[S]=i\}|}{|OU|}$

F_{top}, F_{an} are specified by data publisher

Although the increase in the common nodes of successors of v and u increases data utility, it increases the probability of relationship disclosure. Because this contrasts with privacy requirements, common successor nodes are not considered in UM so that they do not negate the effect of PM. Consider two sets OV and OU which contain nodes that are successors of only v and only u, respectively. If nodes v and u are grouped together, the average distance from each node ($v(u)$) to nodes that are only successors of other node ($OU\ (OV)$)($\frac{\sum_{x\in OU} dist(v,x) + \sum_{x\in OV} dist(u,x)}{|OU|+|OV|}$) is used as a metric to preserve topological properties. The ideal value for this is 1. Since a lower average is more desirable, the inverse of this fraction is used for UM. If the numerator or denominator of this fraction is zero, consider 0.5 for its inverse. In [Fig. 1], consider *Dave* as v and *Fred* as u. Then, OV = {*Alice, Ed*}, OU = {*Fred,Gerg*} and

the inverse of this fraction equals:
$\frac{2+2}{dist(Dave,Fred)+dist(Dave,Gerg)+dist(Fred,Alice)+dist(Fred,Ed)} = \frac{4}{2+3+3+3} = \frac{4}{11}$.

Wave hedges metric [Cha 2007] is used to compare the difference in distribution of sensitive values in the OV and OU nodes. To compute their probability density functions (pdf), we define *SValSet*, which contains all sensitive labels of the OV and OU nodes. For each value $i \in SValSet$, we compute its probability in OV and OU as $p_v(i)$and $p_u(i)$, respectively. Higher values of $\frac{|p_v(i)-p_u(i)|}{max(p_v(i),p_u(i))}$ show a greater difference between the probability of the existence of *i* in OV and OU relative to their maximum. Value of this fraction falls in the range [0,1]. $\sum_{i\in SValSet}\frac{|p_v(i)-p_u(i)|}{max(p_v(i),p_u(i))}$ is the sum of differences of probabilities for all values of *SValSet*. To normalize the difference, it is divided by |*SValSet*|. This value is in range [0,1] and indicates the difference between two pdfs. Since more similarity is desirable for UM, subtract it from 1. In the example, for *Dave* and *Fred, SValSet*={3500,7000,5000}, $p_{Dave}(3500) = \frac{1}{2}$ and $p_{Fred}(3500) = 0$. This produces: $1 - \frac{\sum_{i\in SValSet}\frac{|p_v(i)-p_u(i)|}{max(p_v(i),p_u(i))}}{|SValSet|} = 1 - \frac{\frac{|1/2-0|}{1/2}+\frac{|1/2-1/2|}{1/2}+\frac{|0-1/2|}{1/2}}{3} = \frac{1}{3}$.

F_{top} and F_{an} are coefficients that show the importance of preserving the utility of analysis. They have important effects on UM values and grouping methods.

As stated in Definition 5, calculation of the first and second terms of UM takes $O(|OV| + |OU|)$ and $O(|SValSet|)$, respectively; both time complexities are less than $O(v[dout] + u[dout])$. So time complexity of UM for all pairs is $\sum_{v\in V}\sum_{u\in V}(v[dout] + u[dout]) \in O(n^3)$.

[Section 4] applies a greedy algorithm for grouping individuals. This algorithm creates groups with a minimum number of members and maximum average weight for all pairs.

4 Anonymization Algorithm

This section describes greedy algorithm grouping social-network data (*GroupingSND* (Grouping Social-Network Data) based on the DM for implementing ASN technique. The algorithm partitions individuals of network data *N* into non-overlapping groups so that each group satisfies privacy requirements of (α,β,γ,δ)-SNP. To decrease information loss, construct groups in small sizes. First, compute the DM for all pairs of individuals and then apply a greedy algorithm to group them. This algorithm makes groups with a minimum number of members and maximum average desirability.

[Fig.3] depicts the algorithm. First priority was assigned to all nodes (individuals) (lines 1-3). Priority *v* indicates the similarity of individual *v* with other individuals in network data *N* for tabular (quasi-identifiers and sensitive) values and structural (in-degree, out-degree, relationship) properties. The priority of each node is calculated as:

$$priority[v] = \sum_{x \in \{QI_1,\dots,QI_q,S,Din,Dout\}} \left(F_X \frac{\pi_{count(*)}\left(\sigma_{x=v[x]}(T)\right)}{n} \right) + F_{succ} \frac{\sum_{w \in v[succ]} (w[Din] - 1)}{n} \quad (4)$$

F_x and F_{succ} are coefficients represented in PM (F_x equals F_{QI} for $x \in \{QI_1, \dots, QI_q\}$). The rate of property x for an individual is calculated using the number of individuals with the same value $v[x]$ divided by the total number of individuals in network data N. In the dataset shown in [Fig. 1], the rates for the zip code for *Alice* and *Bob* are $\frac{2}{7}$ and $\frac{1}{7}$, respectively because there are 2 tuples with *zip code* 23000 and 1 tuple with 21000. To measure the similarity of successor nodes, compute the sum of the in-degrees of all successor nodes of current individual v. For each $w \in v[succ]$, there are ($w[Din]$-1) other individuals (besides v) with successors sets having an intersection with $v[succ]$ (at least w is a common member). If individuals v and u have more than one common successor, u is counted for each common successor node in $\sum_{w \in v[succ]} (w[Din] - 1)$, because the number of common nodes in the successor sets of individuals increases their similarity. Finally, the priority of v is the sum of similarity rates of all quasi-identifiers, sensitive attribute, in-degree, out-degree, and successor nodes of individual v.

Next, all nodes are sorted in descending order based on their priority and put in the unmarked list (line 4). Early determination of group members with more similarity (higher priority) has an important impact on group size. The goal is to group vertices with more differences in their properties, most individuals with the least similarity (low priority value) are selected first. In the end, only similar individuals with high priority remain ungrouped. This causes the group size of remaining individuals to grow. In each step k, the first member of group k is the node with maximum priority in the unmarked list (lines 8-9) and it is removed from the unmarked list (line 10). In this way, the chance of finding unmarked vertices with a greater number of different properties and the probability of belonging to a smaller group size grows.

Next, other members of group k are determined (lines 11-18). The node with the maximum average DM with the current members of group k ($\frac{1}{|G_k|} \sum_{\forall r \in G_k} DM[v_t, v_r]$) is selected, removed from the unmarked list, and added to members of group k (lines 12-14). If there are multiple nodes with the same DM average, the node with the maximum priority is selected. The privacy parameters for the current group k are then measured (line 11). If it does not satisfy the privacy requirements of (α,β,γ,δ)-*SNP* with current members, the next node from the unmarked set is similarly selected; otherwise, the next step *k+1* is serviced (line 7).

Generation of groups continues until the unmarked set becomes empty. Finally, in lines 17-24, if the last group does not satisfy (α,β,γ,δ)-*SNP*, delete it and add its nodes to the previously-generated groups. For each node in this last group, find the group with the maximum DM average so that adding this new node does not violate (α,β,γ,δ)-*SNP* privacy requirements.

Algorithm: GroupingSND
Input: N=(V,E) where |V|=n and parameters α, β, γ and δ, matrix DM[n][n] contains DM value of all pairs
Output: QID-groups
1) for i← 1 to n do
2) calculate *priority*[v_i] based on equation 4
3) end for
4) *unmarked* ← Sort-descending(*priority*, *V*) ; //all nodes put on unmarked list
5) $k \leftarrow 0$; // group ID
6) repeat
7) $k \leftarrow k + 1$; // generate next group
8) $i \leftarrow$ first node of unmarked list; //node with $\max_{\forall j \in unmark} priority[j]$
9) $G_k \leftarrow \{i\}$; //i is the first member of group k
10) *unmarked* ← unmarked – {*i*}; //remove i from unmarked list
11) while (G_k *dose* not satisfy privacy requirments) and *unmarked* ≠ ∅ do
12) $i \leftarrow$ individual with $\max_{\forall t \in unmark} \frac{1}{|G_k|} \sum_{\forall r \in G_k} DM[v_t, v_r]$ //next member
13) $G_k \leftarrow G_k \cup \{i\}$
14) *unmarked* ← *unmarked* – {*i*}
15) end while
16) until *unmarked* = ∅ //stop generate group when no ungrouped node
17) if (G_k *dose* not satisfy privacy requirments) //if last group is incomplete
18) for each $q \in G_k$
19) $G_j \leftarrow$ group with $\max_{1 \le j < k} \frac{1}{|G_j|} \sum_{\forall r \in G_j} DM[v_q, v_r]$ adding q does not violate privacy constraints of G_j
20) $G_j \leftarrow G_j \cup \{q\}$ // add memebers of last group to other group
21) $G_k \leftarrow G_k - \{q\}$
22) end for
23) $k \leftarrow k$ -1 //last group removed
24) end if
25) return $G_1, \ldots, G_k$

Figure 3: Greedy algorithm to generate groups

The number of repetitions of inner instruction (lines 11-18) is $O(n)$. In each execution of instructions, one vertex is removed from the unmarked list until all vertices are selected. In each iteration of the while loop, line 12 takes $O(|G_k| * |unmark|)$. The high bound for it is $O(|G| * n)$, where $|G|$ is the maximum group size. The time complexity for computing sensitive and degree disclosure probabilities are $O(1)$. The time complexity of computing presence and relationship disclosure probabilities (to count all *PWs* in the worst case) equal $|G|!$. The complexity of the function *privacy_req* is $O(|G|!)$. The upper bound for lines 6-16 is $O\big(n * (|G|! + |G| * n)\big)$. Lines 17-24 are related to the last group and the complexity of line 19 is $O(n)$; making the total time for these lines $O(|last\ Group|.n)$. The time complexity for computing the priority of each individual u is $O(n * q + u[Dout])$. Lines 1-3 are $O(n^2 * q)$. Sorting the list in line 4 is $O(n * \log n)$. Therefore, the overall time cost of the algorithm in the worst case is $O(n^2 * q + n * \log n +$

$n * |G|! + |G| * n^2 + |last\ Group| * n)$. If $(|G| - 1)! < n$, the time complexity is $O(|G| * n^2)$; otherwise it is $O(n * |G|!)$. The anonymization algorithm in data publishing is an offline algorithm and its time complexity is not critical.

5 Measurement of Information Loss

In the proposed anonymization algorithm, the groups were generated so that all their members satisfy the requirements of the privacy model. To evaluate the algorithm, we measure the information loss of the anonymized network data that it generates.

Let Q be a count query $Q(N)$ (actual result) and let $Q(N^*)$ (anonymized result) be the accurate and approximate results by applying Q to original network data N and the released network data N^*, respectively. The relative error equals the proportion of absolute difference of the actual and anonymized results to the actual result, in this way the small difference increases more the relative error when the actual value is small with respect to its large value:

$$E = \frac{|Q(N) - Q(N^*)|}{|Q(N)|} \quad (5)$$

As in [Wang 2010], [Xiao 2006], this metric is used to compute information loss. Four types of queries were used to measure information loss of the proposed anonymization algorithm as follows:

5.1 Aggregate tabular query

The relationship between individuals is not considered for queries on tabular data. Assume that each individual has some attributes (quasi-identifiers, sensitive attribute, in-degree and out-degree). Each query of this kind is count query $Q = count(\sigma_C(N))$. It can be transformed to $Q = count(\sigma_C(AT_1 \bowtie AT_2 \bowtie DT \bowtie ST))$ on the released ASN schema $(AT_1, AT_2, ST, DT, SVT)$, where C is a selection condition. There is no foreign key between these tables, so many false tuples are generated by the lossy join of the tables $(AT_1,\ AT_2,\ ST, DT)$. Query Q on their lossy join produces more tuples than the original tuples in N; thus, estimate the result of query Q using the following method:

Approximate ASN estimates $Q(N^*)$ by applying an estimation to tables AT_i, ST and DT. Use $C_i (1 \le i \le 2)$, C_d and C_S to denote the results of applying selection condition C on the schema of tables AT_i, DT and ST. If the selection condition does not contain attributes of any of the above tables, the condition for that table would be empty. For example, for C = '*job* = nurse and *Din* = 1 and *income* = 3500 on the *ASN* scheme in [Fig. 2], C_1 (on AT_1) ='*job = nurse'*, C_2(on AT_2)='', C_d(on DT) = '*Din = 1'* and C_S (on ST) = '*Disease* = stroke'. Furthermore, each condition can take two forms: equality or range. The range condition is in the form (x_1< attribute <x_2) for numerical attributes and the attribute in $\{x_1, x_2, \ldots, x_v\}$ for non-numerical or categorized attributes such as *job*.

The pseudo code in [Fig. 4] shows the details of how to approximate the result of the count queries. First, determine all groups that satisfy C_S (line 1). Second, for each group G_j, estimate the count result (lines 3-15). In particular, compute count result s_j as the sum of *count* attribute that satisfies C_S in sensitive table ST where GID = j (line 4). Then, for every selection condition C_i on table AT_i $(1 \le i \le 2)$ (and C_d) calculate

probability p_i (and p_d) of the tuples in G_j that satisfy C_i (and C_d). The result of multiplication of these probabilities is stored in p. Each tuple in G_j satisfies all conditions C_1, C_2, and C_d with probability p (lines 7-16). Then adjust the count result accordingly by multiplying s by p (line 17). Finally, sum the adjusted counts for all groups (line 17). This sum is an estimation of the result of query Q for anonymized data based on *ASN*.

Algorithm: estimateAggregateTabularQuery
Input: *ASN* tables(AT_1,AT_2,ST,DT,SVT), query Q
Output: the estimated result of Q

```
1)  GIDS ←Π_GID (σ_Cs(ST)); //groups that satisfy C_s
2)  n ← 0
3)  for each group ID j Є GIDS do
4)      s_j←Π_sun(count)(σ_(Cs,GID=j)(ST)); //memebers in G_j that satisfy C_s
5)      |G_j|←Π_sun(count)(σ_GID=j(ST)); // compute group size
6)      p ← 1;
7)      for each C_i isn't empty do
8)          k ←Π_sun(count)(σ_(Ci,GID=j)(AT_i)); //memebers in G_j that satisfy C_i
9)          p_i←(k/|G_j|);
10)         p ← p × p_i;
11)     end for
12)     if C_d isn't empty then
13)         k ←Π_sun(count)(σ_(Cd,GID=j)(DT)); //memebers in G_j that satisfy C_d
14)         p_d←(k/|G_j|);
15)         p ← p × p_d;
16)     end if
17)     n ← n + s_j × p;
18) end for
19) return n;
```

Figure 4: aggregate tabular query estimation algorithm

Example: Using the *ASN* scheme in [Fig. 2], show how the algorithm in [Fig. 4] operates on the following query to estimate the results of count queries for query Q_1:

SELECT count(*)
FROM Released-network-data
WHERE job ="nurse" AND Din=1 AND income=3500;

Only group 1 satisfies the condition *income* = 3500 on *ST* ($S_1 = 1$). For group 1, $p = \frac{2}{4} \times \frac{3}{4} = \frac{6}{16}$, where $p_1 = \frac{2}{4}$ corresponds to 2 tuples out of 4 in group 1 that satisfy *job = "nurse"* in table AT_1, and $p_d = \frac{3}{4}$ corresponds to 3 tuples out of 4 in group 1 that satisfy *Din=1* in table DT. The result of this query is $S_1 \times p = \frac{6}{16}$. The result of this query on original network data in [Fig. 1] is 1, since only *Alice* has those conditions. As a result, the relative error is $\frac{\left|\frac{6}{16}-1\right|}{1} = 0.62$.

5.2 Aggregate network query

An aggregate network query calculates aggregation on paths or subgraphs satisfying some query conditions. One example is the average distance from a nurse node to a teacher node in a network structure. For this kind of query, conditions C_{source} and C_{dest} exist for source and destination nodes, respectively. These conditions are defined for attributes of individuals, such as quasi-identifiers, sensitive attributes, in-degree, or out-degree. To execute this kind of query on original network data, first determine the set of all individuals that satisfy those conditions ($SN = \sigma_{C_{source}}(N)$, $DN = \sigma_{C_{dest}}(N)$). Then compute the minimum distance from each node in *SN* to each node in *DN*. Then calculate average ($query\_result = \frac{1}{|SN|\times|DN|}\sum_{i\in SN}\sum_{j\in DN} dist_N(i,j)$), where $dist_N(i,j)$ is the length of the shortest path from *i* to *j* in network *N*.

Next, estimate the result of this kind of query on anonymized network data by *ASN*. As done previously, each condition C_{source} and C_{dest} are divided by related conditions in tables (AT_1, AT_2, ST, DT). Since false individuals are generated by the lossy join of the *AT* tables, there may be tuples of tables (AT_1, AT_2, ST, DT) for G_j that satisfy $C_{source}(C_{dest})$. While, in reality, no members of G_j satisfy all conditions. For every label in each group that satisfies $C_{source}(C_{dest})$, measure the probability that its corresponding individual satisfies $C_{source}(C_{dest})$ in the original data based on the algorithm in [Fig. 5].

Algorithm: estimateLabelwithProbability
Input: *ASN* tables(AT_1,AT_2,ST,DT,SVT), condition $C=\{C_1, C_2, C_s, C_d\}$
Output: the set of pair (label,probability)

1) *GIDS* ←$\Pi_{GID}(\sigma_{Cs}(ST))$; //groups that satisfy C_s
2) *Pairs* ← {};
3) for each group ID *j* ∈ *GIDS* do
4) $s_j \leftarrow \Pi_{sun(count)}(\sigma_{(Cs,GID=j)}(ST))$; //memebers in G_j that satisfy C_s
5) $|G_j| \leftarrow \Pi_{sun(count)}(\sigma_{GID=j}(ST))$; // compute group size
6) *prob* ← $\frac{s_j}{|G_j|}$; //probabilty of each member G_j satisfy C_s
7) for each C_i isn't empty do
8) $k \leftarrow \Pi_{sun(count)}(\sigma_{(Ci,GID=j)}(AT_i))$;
9) $p_i \leftarrow (k/|G_j|)$; //probabilty of each member G_j satisfy C_i
10) *prob* ← *prob* × p_i;
11) end for
12) if *prob*>0 then
13) *labels* ←$\Pi_{label}(\sigma_{(Cd,GID=j)}(DT))$;
14) end if
15) for each *L* in *labels* do
16) *pairs*← *pairs* ∪ (*L*, *prob*);
17) end for
18) end for
19) return pairs;

Figure 5: Algorithm to estimate probability of labels that satisfy condition C

The inputs of this algorithm are ASN tables and the set of conditions on each table (some conditions may be empty). For each group G_j, compute the probability of each tuple in the lossy join of (AT_1, AT_2, ST) for G_j that satisfies conditions $\{C_1, C_2, C_s\}$ (lines 1-11). For each table, calculate the sum of values of the *count* column of tuples in G_j that satisfy the condition (lines 4 and 8) divided by the size of members in G_j (lines 6 and 9) and multiply all these results (line 10). This value (*prob* in [Fig. 5]) shows the probability of each tuple of G_j in the lossy join of those tables that satisfy conditions $\{C_1, C_2, C_s\}$. If this probability is zero for G_j, there is no node in G_j that satisfies C; otherwise all labels in G_j that satisfy C_d may meet all conditions of C with probability *prob* (lines 12-14). For each of those labels, the pair (label, prob) is added to the result (lines 15-17).

To estimate this kind of query for the *ASN* data, extract all labels that satisfy $C_{source}(C_{dest})$ based on the algorithm [Fig. 5] and put them into set $pairs_S$ ($piars_D$). To execute this query, compute the distance from each label in $pairs_S$ to each label in $pairs_D$, but, based on [section 2], for each group there are probable world edges to reconstruct the output edges of each group. If $|PWE_j|$ denotes the number of valid output edges sets for G_j, there are $\prod_{j=1}^{n'} |PWE_j|$ choices to reconstruct all edges of the network. Some of these reconstructed networks are randomly generated. The average of the query result on all randomly reconstructed networks was considered as estimated result of query for the published network data. The estimated result of the query for each reconstructed network *SG* is computed as the weighted average of the distance of all $\{(label[i], label[j]) | i \in pairs_S, j \in pairs_D\}$. The weight of each distance of each pair $(label[i], label[j])$ is $prob[i] \times prob[j]$. The query result on network *SG* is:

$$\frac{1}{\sum_{i \in pairs_S} \sum_{j \in pairs_D} (prob[i].prob[j])} \times \sum_{i \in pairs_S} \sum_{j \in pairs_D} (prob[i].prob[j].dist_{SG}(label[i], label[j])) \quad (6)$$

5.3 Graph Topological Properties

One of the most important applications of social network data is analysis of graph properties. To understand and utilize the information in a network, various measures have been developed to describe the structure and characteristics of the network from different perspectives. Some of these measures are degree sequences, shortest connecting paths, clustering coefficients, closeness and betweenness. As stated above, there are multiple candidate graphs to reconstruct the network from the released data based on ASN. As for the previous query, evaluate the structural measures on some reconstructed graphs and then compare the averages of their properties with the original network. The anonymization algorithm has no effect on degree sequence because, in *ASN* technique, the degree of nodes remains unchanged; we consider the following measures:

- *Connectedness*: Every anonymization algorithm can modify the connectivity of the network (split a component or combine multiple components). Since the network graph is assumed to be directed, consider this measure in two ways:

  - *Size of maximum strongly-connected component*: (strongly connected component with maximum nodes)
  - *Size of maximum weakly-connected component:* (weakly connected component with maximum nodes)
- *Shortest path length*: we evaluate the effect of the anonymization algorithm on the shortest path lengths in the graph. Since it is possible to have no path from one node to another, consider three cases to compare shortest paths (because the network structure is directed, there are $n(n-1)$ pairs to measure their shortest path):
  - *Existing paths*: Compute the average distance of all pairs having a path from source to destination
  - *All pairs*: Compute the average path length between all pairs; for pairs with no path, let the path length be $2|V|$ instead of an infinite value
  - *Selected pairs*: Select 100 random pairs (source node, destination node) having paths between them in the original network and compute average distance of them
- *Diameter*: The original directed network may not be strongly connected. To neglect infinite paths, consider the maximum shortest path lengths of all existing paths as the diameter.
- *Closeness*: Closeness of a vertex v is $\frac{1}{\sum_{t \in V} dist(v,t)}$. If there is no path from v to t, consider the distance as $2|V|$.
- *Betweenness*: This quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It is introduced as a measure to quantify the importance of an individual in communication between others in a social network.
- *Clustering coefficient*: It is a measure of the degree to which nodes in a graph tend to cluster together. To compute this scale, the edge direction is ignored. If n_v is the set of all neighbors of v, the clustering coefficient of v is the number of pairs in n_v that are adjacent to each other in the network divided by the number of possible pairs ($|n_v|(|n_v|-1)/2$).

5.4 Graph spectral properties

The graph spectrum has close relations with many graph characteristics and can provide global measures for some network properties. We consider the following metrics:

- *Normalized eigenvector*: The eigenvector is a non-zero vector $v = \{v_1, \dots, v_n\}$; when graph adjacency matrix $A_{n \times n}$ is multiplied by v, it yields a constant multiple of v. The latter multiplier is denoted by λ ($Av = \lambda v$) [Ying 2008]. The normalized eigenvector equals $nv = \frac{1}{\sum_{i=1}^{n} v_i} v$.
- *Page rank:* The page rank of each vertex specifies a score that is the fraction of time spent visiting that vertex (measured over all time) in a random walk over the vertices (following outgoing edges from each vertex) of the graph. Compare the page rank of each vertex in original and anonymized network.

- *Spearman similarity of top k ranked vertices:* Higher ranked vertices have more effect on preserving graph utility. Sort the vertices of the graph based on their page rank scores, then extract the top 5% ($k = 0.05n$) of them in ranked lists L and L^* for the original and anonymized graph. As [Milani Fard 2013], evaluate the Spearman similarity (SS) between these lists as $SS = 1 - \frac{2(k-|Z|)(K+!)+\sum_{i\in Z}|r_L(i)-r_{L^*}(i)|-\sum_{i\in S} r_L(i)-\sum_{i\in T} r_{L^*}(i)}{k(k+1)}$, where Z is the set of nodes in both L and L^*, S is the set of nodes that are only in L, T is the set of nodes that are only in L^*, and $r_L(i)$ is the rank of node i in list L. The range of SS is [0,1] where 0 denotes totally reserved and 1 denotes totally identical.

6 Experimental Results

Experiments were conducted on datasets to evaluate information loss of the anonymized network data generated by our *GroupingSND* based on *ASN* technique. We compared the results of the four types of queries in [section 5] for the anonymized and original networks. The complete comparison of the proposed algorithms with other anonymization methods for all four types of queries is impossible for the following reasons:

1. There is no privacy model other than (α,β,γ,δ)*-SNP* to protect against *four* kinds of disclosure of private information (*membership disclosure, sensitive attribute disclosure, degree disclosure, relationship disclosure)* on the directed social network. Existing research only protects against some of these attacks. Providing more privacy protection decreases utility [Wu 2009].
2. The social network model is considered to be an undirected graph in almost all existing research [Bhagat 2009], [Campan 2008], [Hay 2010b], [Liu 2008], [Tai 2014], [Zhou 2011], but a directed graph for the proposed privacy preserving method.

Despite this, we compared the proposed algorithms with Subgraph-wise Perturbation (SP) [Milani Fard 2012] and Neighborhood Randomization (NR) [Milani Fard 2013] for the graph topological properties, although they only protect against relationship disclosure in the directed graph. In SP and NR, quasi-identifiers and sensitive attributes for nodes are not considered. Since the goal of NR is not to protect re-identification, it does not consider any background knowledge, such as degree.

6.1 Experimental Setup and Datasets

Setup: We implemented the proposed anonymization algorithm in JAVA and used JUNG 2.0.1[1] software library to manipulate and analyze the graphical data. The used system platform was Windows 7, Oracle 11g, Intel core i4 with 2.4GH and 14G RAM.

Datasets: We evaluated ASN and GroupingSND on the following network data:

- Wikivote[2] which contains all Wikipedia voting data from the inception of Wikipedia until January 2008. Nodes in the network represent Wikipedia users. A directed edge from node *i* to node *j* shows that user *i* has voted for user *j*. The network contains 7115 nodes and 103689 directed edges.

- URVEmail[3] which contains edges of e-mail interchanges between members of the University of Rovirai Virgili (Tarragona). This email network contains 1133 nodes and 10933 directed edges.
- Random, which was generated by Pajek[4] with 2000 nodes. It contains 109832 edges.

Since the nodes should contain quasi-identifiers and sensitive labels, micro-dataset Census [5] was used, which contains personal information of 1,000,000 Americans. Census was produced by the data extraction system of the US Bureau of Census. This dataset contains 7 QID-attributes and one sensitive attribute. Details of the attributes are summarized in [Tab. 1]. Random sets of 7115, 1133, and 2000 tuples were assigned to the above networks nodes.

Attribute	Age	Gender	Maritial	Race	Birth Place	Education	Work class	Salary
Number of distinct value	100	2	6	9	144	12	16	950

Table 1: Summary of attributes of dataset

	D(indegree)	D(outDegree)	D(sensitive)	Indegree range	Outdegree range
Wikivote	0.66	0.335	0.235	0-457	0-893
URVEmail	0.132	0.132	0.228	1-71	1-71
Random	0.059	0.059	0.17	26-83	30-84

Table 2: properties of datasets

[Tab. 2] shows details of each dataset. The function *D(x)* (density of property *x*) returns the frequency of the most frequent value for attribute *x* in network data *N* divided by the number of all individuals in *N*. As mentioned at the end of [Section 3.1], to find a grouping were all groups satisfy privacy requirements, the thresholds of β, γ, and δ should be more than the density of the related properties. In the worst case, if all individuals are located in one group, this group should satisfy privacy constraints. We set privacy parameters for all experiments to $\alpha = 0.25$, $\beta = 0.25$, $\gamma = 0.7$, $\delta = 0.7$, $F_{top} = F_{an} = 4$.

Wikivote has power law degree distribution [6]. In this dataset, more than half of the nodes have out-degrees of 0 or 1 and in-degree 0; a few nodes have out-degrees greater than 400. In this dataset, finding a grouping based on *ASN* such that all its groups satisfy relationship privacy is difficult. In the proposed algorithm, if current members of G_k do not satisfy the δ-relationship constraint, next node added to G_k to decrease the probability of relationship disclosure for G_k. Adding a node with out-degree 0 to G_k does not increase the output edges of G_k; thus, its probability of relationship disclosure remains unchanged. There are many nodes with this property and all of them should be located in groups. This increases group size. The size of the group that includes the nodes with the highest out-degree (893) is large. To satisfy the δ-relationship of that group, the nodes with high degree having few intersections with successor nodes should be added to it. A few nodes have high out-degree and most of them are connected to nodes with high in-degree (intersection at successor nodes), which increases the size of this group.

In the Random dataset, the range of in-degrees and out-degrees of all nodes is limited; therefore, the degrees of nodes are close to each other. Here, finding a grouping to satisfy the degree and relationship privacy is easy.

URVEmail has a power law degree distribution, but the range of degree is more limited than for the Wikivote dataset. Finding the *ASN* groups is not as difficult as for the Wikivote dataset.

Since satisfying privacy constraints differs from one dataset to another, the values of $F_{QI}, F_S, F_{Din}, F_{Dout}$ and F_{succ} were different for each dataset. For example, in Wikivote, the density of in-degree is near its constraint of $\gamma = 0.7$ and satisfying relationship privacy is difficult; thus, F_{Din} and F_{succ} should be higher than other coefficients.

6.2 Results

The proposed algorithm was evaluated for the network types described previously. To measure information loss, queries were generated in each of the four types [section 5] and the relative error of results of queries on anonymized network was computed for comparison with the original network data.

[Fig. 6(a)] shows the minimum, average, and maximum group size for each set of anonymized data. Group size has an important impact on information loss (a decrease in group size decreases information loss). Because β equals 0.25, the minimum possible group size to satisfy the sensitive association requirement is 4. [Fig. 6(a)] shows that there were groups with 4 members in all experiments. The average and maximum group size for the URVEmail and Random datasets were similar and close to each other. The average was close to the minimum group size. In Wikivote, the average and maximum group size were greater than for other datasets because it was difficult to attain the degree and relationship requirements for groups with members having high out-degrees.

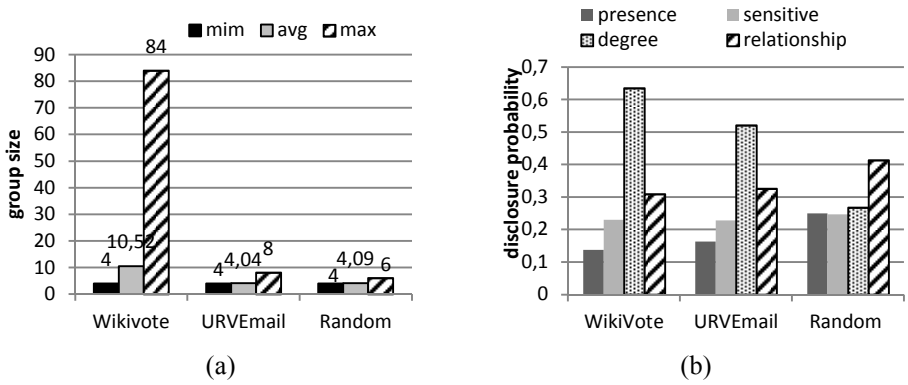


Figure 6: (a) minimum, average and maximum group size, (b) average of individuals privacy disclosure probability

Each individual is a member of one group, but the probability of privacy disclosure of each group may differ from other groups. [Fig.6(b)] illustrates the

average probability of privacy disclosure for all network individuals. As shown, all disclosure probabilities were below their specified thresholds.

6.2.1 Aggregate Tabular Query

Two types of aggregate tabular queries were considered: (1) queries with only equality conditions and (2) queries with range conditions for quasi-identifiers, in-degree, and out-degree. In both types, the condition for sensitive attribute was equality. To evaluate each anonymized network, 100 equality queries and 100 range queries were randomly generated. For each random query, properties (quasi-identifiers, in-degree, and out degree) were chosen randomly and a random condition was created for each property. The number of conditions differed from one query to another. The selectivity of one query is defined by the number of individuals satisfying all its conditions. Increasing the number of selected properties decreases selectivity. In addition, selectivity of range queries is more than equality queries with the same selected properties.

For evaluation, the relative error (equation 5) was computed for each random query. [Fig.7] shows the minimum, average, and maximum relative error for all random queries for each dataset. As shown, the average of relative error for range queries was lower than that for equality queries in all datasets. The average error for all datasets was close to each other for equality queries.

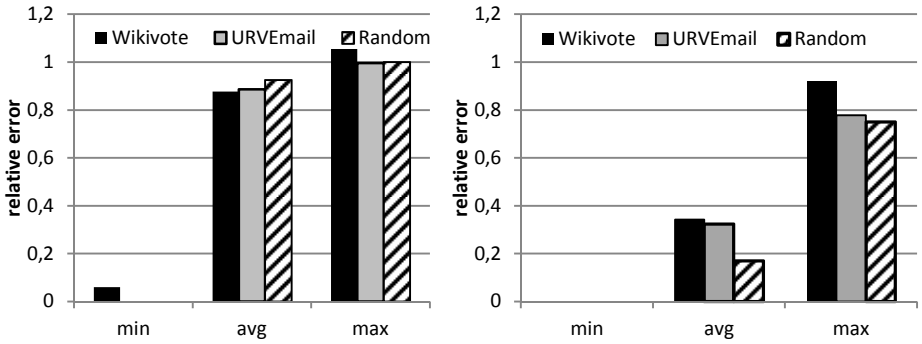
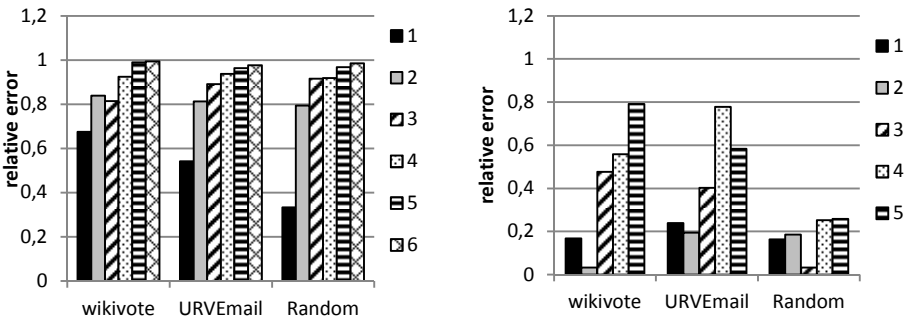


Figure 7: relative error for aggregate tabular queries

[Fig.8] shows the average relative error by the number of properties for the query conditions for all experiments. It shows that, in both queries, accuracy decreased when the number of properties increased for the query condition. As mentioned, increasing the number of properties in query condition usually decreases selectivity.



(a) equality queries (b)) range queries

Figure 8: relative error of aggregate tabular queries with respect to number of involved properties in query conditions

6.2.2 Aggregate Network Query

In this kind of query, the goal is to compute the average distance from source nodes with specified properties to destination nodes with determined properties for all random reconstructed graphs. For each dataset, 100 random queries of this kind were generated. To describe the source nodes for each random query, properties (quasi-identifiers, in-degree, and out degree, sensitive) were randomly chosen to create random conditions for each property. Condition of each property was able to be either equality or range condition. The same was done to describe the properties of the destination nodes.

[Fig.9(a)] represents the minimum, average and maximum relative error of all generated queries in all experiments. As shown, information loss from the Random dataset was lowest. Although the average error of all datasets was relatively small, the maximum relative error for Wikivote was high. Since the size of some groups for the Wikivote dataset was large, the number of false combinations generated by lossy join increased. This increased the information loss for some queries in those groups.

[Fig.9(b)] shows the average error for all datasets by the number of properties involved in source and destination conditions. Since each condition of query was able to be either equality or range, there was no general trend for increasing the number of properties involved. Since a query with more conditions can have more range conditions, its selectivity can increase.

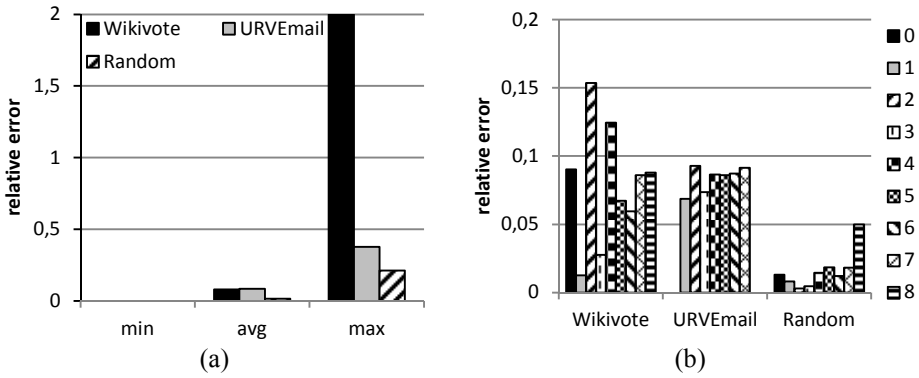


Figure 9: relative error on aggregate network queries: (a) minimum, average, and maximum relative error, (b) average relative error based on number of involved properties in query conditions

6.2.3 Graph Topological Properties

To evaluate information loss for topological properties, the first step is to reconstruct the graph from released tables. The graph edges should be reconstructed based on tables DT and SVT. As mentioned in [section.2], for each group *j, Dout* in DT indicates how many times each label appears as the source of an edge and *count* in SVT shows how many times each label appears as a destination edge for that group's members. There are *probable world edges* (PWE_j) for reconstructing the output edges of G_j; therefore, there are $\prod_{j=1}^{n'} |PWE_j|$ choices to reconstruct the graph edges. For each dataset, 100 reconstructed graphs were randomly generated. In each reconstructed graph, for each group G_j, one member of PWE_j was generated randomly. Next, the topological properties explained in [section 5.3] were measured for each reconstructed and original network and the average of each property on all reconstructed networks was compared with the original one [Tab.3]. The topological properties of our method in most cases were very close to actual values in the original network. The size of the largest strongly-connected component and largest weakly-connected component changed for Wikivote. The changes in the clustering coefficients for URVEmail and Wikivote were considerable.

If only the average of a property (such as betweenness) on all nodes in one reconstructed network is compared with the original network, then $EA = \frac{|\text{avg}_{v\in V} Q(v) - \text{avg}_{v\in V} Q(v^*)|}{\text{avg}_{v\in V} Q(v)}$($v^*$denotes the corresponding v in N*). This is not a good measure for evaluating the anonymization algorithm, because, when betweenness of one node decreases 10 units and another node increases 10 units, the average remains unchanged while it creates information loss for the betweenness. On the other hand, the value of a property such as betweenness or the clustering coefficient can be zero for some nodes in the original network. This means it is not reasonable to use the average relative error of all nodes $AE = \frac{\sum_{v\in V} \frac{|Q(v) - Q(v^*)|}{Q(v)}}{|V|}$ (division by zero occurs).

Instead, an additional measure was considered to provide better evaluation than *EA*. The proportion of the average absolute difference by the average of all nodes values as calculated by $EDA = \frac{\text{avg}_{v \in V} |Q(v) - Q(v^*)|}{\text{avg}_{v \in V}\, Q(v)}$ was used.

		Strongly component	weakly component	existing shortest path	Selected shortest path	diameter	Closeness	Clustering coefficient	Betweenness	Spearman similarity
Wikivote	Original	1300	7066	3.48	2.92	10	0.0000849	0.141	3930.5	1
	GroupingSND	1300.7	7071.4	3.36	2.57	8.36	0.000085	0.0664	3656.5	0.771
URVEmail	Original	1133	1133	3.61	3.74	8	0.282	0.221	2950	1
	GroupingSND	1133	1133	3.33	3.41	7.56	0.305	0.0675	2636.2	0.882
	SP1k	1132.6	1133	3.60	3.61	8.22	0.273	0.0261	2948.3	0.033
	SP50k	1126.1	1132.9	3.63	3.65	8.2	0.102	0.1622	2955.8	0.531
	NR	1095.8	1133	3.48	3.50	7.9	0.012	0.245	2716.5	0.765
Random	Original	2000	2000	2.19	2.21	3	0.45732	0.0543	2374.2	1
	GroupingSND	2000	2000	2.19	2.20	3	0.4527	0.0547	2374.6	0.969

Table 3: topological properties of original and anonymized network

[Fig.10] shows the average relative error on all random reconstructed graphs for all topological and spectrum properties explained in [sections 5.3 and 5.4]. The diameter of all reconstructed graphs was equal or close to the diameter of the original network in all datasets. Wikivote had the highest average relative error for this property.

As explained in [section 5.3], the average distance between all two nodes was computed for each reconstructed graph and original network. In this case, it was possible for there to be no path between some pairs; so the average distance between all pairs with some paths on the network was calculated. [Fig. 10] shows that the values of EA for the Random dataset were close to zero and that URVEmail had the highest relative error. Furthermore, for each original network, 100 pairs with paths between them were selected. Then the AE of shortest path length of each pair in anonymized network was computed. As mentioned, calculation of the EA does not provide a good metric. As seen, the AE of the selected pairs in the Random dataset was greater than that for the URVEmail dataset, while the EA for the Random dataset approached zero.

[Fig.10] shows the EDA for the closeness, betweenness and clustering coefficient since their values for some nodes in the original network could be zero. As shown, the proposed algorithm preserved closeness and betweenness in all datasets very well. It preserved the clustering coefficient for the URVEmail and Wikivote datasets at a satisfactory level.

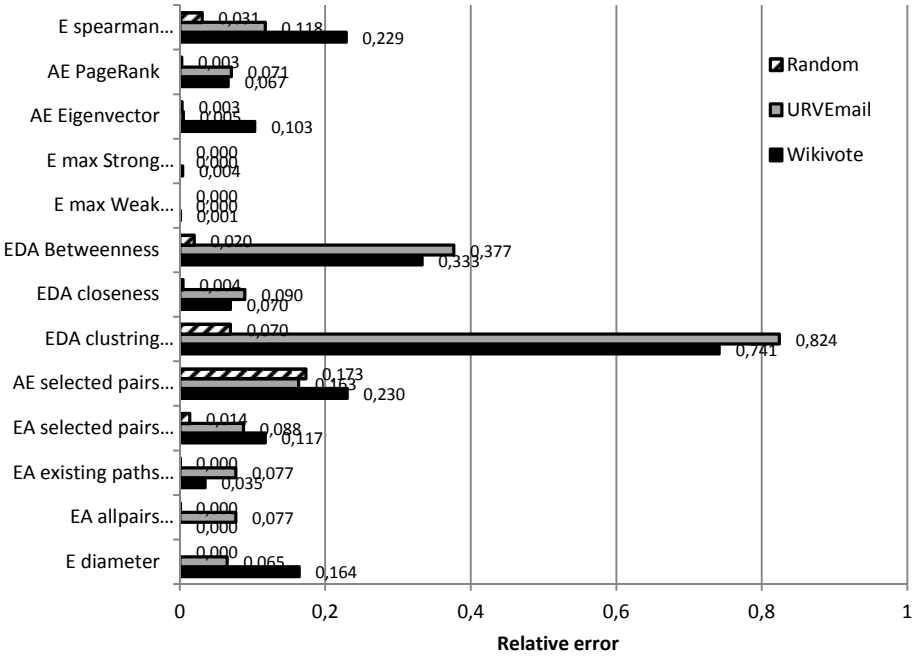


Figure 10: Topological and spectrum properties in original and anonymized network

The maximum strongly-connected component size (MSCCS) and maximum weakly-connected component size (MWCCS) remained unchanged for the Random and URVEmail datasets, but changed in the Wikivote dataset. These changes had an important impact on the diameter and shortest path length.

The eigenvector and page rank vectors were first normalized and the relative error of corresponding elements of the two vectors was computed. Finally, the average relative errors (AE) were computed. As seen, their relative errors were low for all datasets, especially for the Random dataset. To compute the relative error of spearman similarity a value of 1 was considered to be its actual value in the original network. For Wikivote, the spearman similarity showed greater information loss.

6.2.4 Effect of Priority Measure

In the proposed anonymization algorithm, the first member of each group is chosen based on its priority measure. Now in this Section, we consider same priority for all nodes. The URVEmail dataset was tested to evaluate the effect of priority measure. [Fig.11] shows the average of relative error for equal, range and aggregate network queries. As shown, the relative error increased slightly. [Fig. 12] shows the relative error for topological and spectrum properties. As shown, the relative errors were similar. In some case, *GroupingSND* without priority has lesser information loss. As mentioned in [section 4], some nodes remain in the last incomplete group because there are no unmarked nodes to add to this group and because its privacy

requirements are not satisfied using the current members. Without the use of priority, the members remaining in last group increased. Each node should be added to a suitable available group without violating privacy requirements.

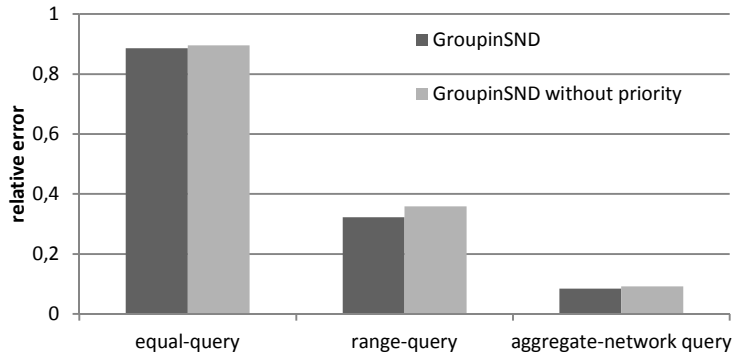


Figure 11: Evaluation of removing priority measure on average of relative error on equal, range and aggregate network query

6.2.5 Summary of Results

Experimental results showed that the relative errors of aggregate network queries and graph topological and spectrum properties were lower than that for aggregate tabular queries. This means that ASN preserves the structural properties better than tabular properties. The proposed anonymization algorithm preserves data utility for all datasets at an acceptable level. A utility metric should be considered for only the tabular data in desirability metric to decrease information loss in tabular queries. On the other hand, for tabular queries that do not consider structural data, the publishing relational data methods introduced in [Fung 2010] can be employed.

Notice that the relative error approaching 1 is not unfavorable (equation 5) in cases where the actual value approaches zero, because a slight change in the anonymized value sharply increases the change in error.

Another way to increase the accuracy of analysis of published data is to publish the average relative error for each topological property and each kind of query. In this way, after extraction of the query response from the anonymized data, the actual range of the query response can be estimated. This may not be applicable to all publishing methods since, in the privacy-preserving data publication problem, the method of data anonymization is assumed to be known by the analyzer, so publishing the average relative error may cause privacy disclosure in some methods. All aspects of this issue will be analyzed in future research.

The relative error of the aggregate network queries and graph topological and spectrum properties in the Random dataset approached zero. As a result, the proposed anonymization algorithm preserves data utility well when the node degree of the graph is distributed within a narrow range.

6.3 Comparison with SP and NR

In this section, we first introduce SP and NR, and compare them with the proposed algorithm for the topological and spectrum properties of the graph.

SP: Subgraph-wise perturbation perturbs the destination of a link to achieve uncertainty of inferring the correct destination. It follows the (p_1,p_2) privacy model (0 < p_1<p_2< 1) that states that if prior belief of the adversary is that node v is the destination of a link is not greater than p_1, then his posterior belief that v is the true destination of a link is not greater than p_2. One drawback of this approach is that it is only limits inference of destination nodes that have low in-degrees (prior belief less than p_1). In contrast to SP, the proposed approach limits the probability of relationship disclosure to less than δ(=p_2) for all edges of the graph. To compare SP with the proposed algorithm we consider $p_1 \geq \frac{\max_{v \in V}\{v[Din]\}}{|\{v \in V|(v,u) \in E\}|}$ so that all nodes satisfy prior belief. The graph is partitioned in SP into link-partitioned subgraphs $G_1,\ldots,G_k$. For each (directed) link (u,v), it is retained with certain probability p and replaces (u,v) with link (u,w) with probability 1-p, where w is randomly-selected from nodes in each subgraph. This method is called SP. A larger k increases retention probability, and decreases the reconstruction error, and increases the threat of identifying a true link [Milani Fard 2012].

NR: Neighborhood randomization also replaces the destination with random node w with probability 1- p, but selects w from a local neighborhood of u. NP protects relationship privacy by ensuring that probability of an observed link in the published graph being a true link is not greater than δ. Two parameters influence information loss of a published graph: (1) radius of neighborhood (r) and (2) size of candidate set for random replacement of destination edges from u (s). Smaller s and r values mean that a randomized destination is chosen from a more compact neighborhood with fewer choices, leading to better preservation of the graph structure. Larger s and r values create more uncertainty for a randomized destination, hiding the true destination better [Milani Fard 2013].

SP and NR only protect against link disclosure. The proposed algorithm protects against four types of disclosure. Since is no better choice, the proposed algorithm was compared with SP and NR. The proposed algorithm has greater information loss for some structural properties for URVEmail dataset than for other datasets [Fig. 10]; this comparison was carried out using URVEmail. In all cases, the maximum disclosure of a link was 0.7. In SP, p_1 and p_2 were set to 0.1 and 0.7, respectively. In NR, r and s were set to 2 and 3, respectively. For SP, experiments with different k values (1 and 50) were run and are denoted by SP1k and SP50k, respectively. Since SP and NR are random algorithms, each was run 10 times. [Tab.3] and [Fig.12] show the average structural properties and information loss for all generated anonymized networks for each experiment. The results of the comparison are:

1. The proposed method (*GroupingSND*) did not change the in-degree and out-degree of nodes in the published graph. SP and NR retained the out degree of nodes, but distortions occurred in the in-degrees of the nodes.
2. Relative errors for MSCCS and MWCCS were very low for all methods [Fig.12], but small changes in MSCCS had an important effect on changes in

the distance between pairs. NR had the highest relative error for MSCCS in comparison with SP and GroupingSND.

3. Relative error for the diameter of all methods was very low and close to each other. GroupingSND had the highest relative error in diameter.
4. NR had the highest error for MSCCS; thus, the EA of the distances of all pairs and the AE of distances of pairs with paths between them were very high. [Fig.12] shows that the relative error of GroupingSND for the EA of the distances of all pairs and the AE of existing paths lengths were much lower than those for SP and NR. The AE of the selected pairs distances were close to each other for all methods.
5. In comparison with SP and NR, GroupingSND preserved closeness well. Information loss from GroupingSND for betweenness was similar to that for SP50*k* and much lower than that of SP1*k*. NR preserved betweenness better than GroupingSND.
6. The EDA of the clustering coefficient for SP50*k* and NR was smaller than that for GroupingSND.
7. GroupingSND preserved the page rank, eigenvector, and Spearmen similarity much better than did the other methods.

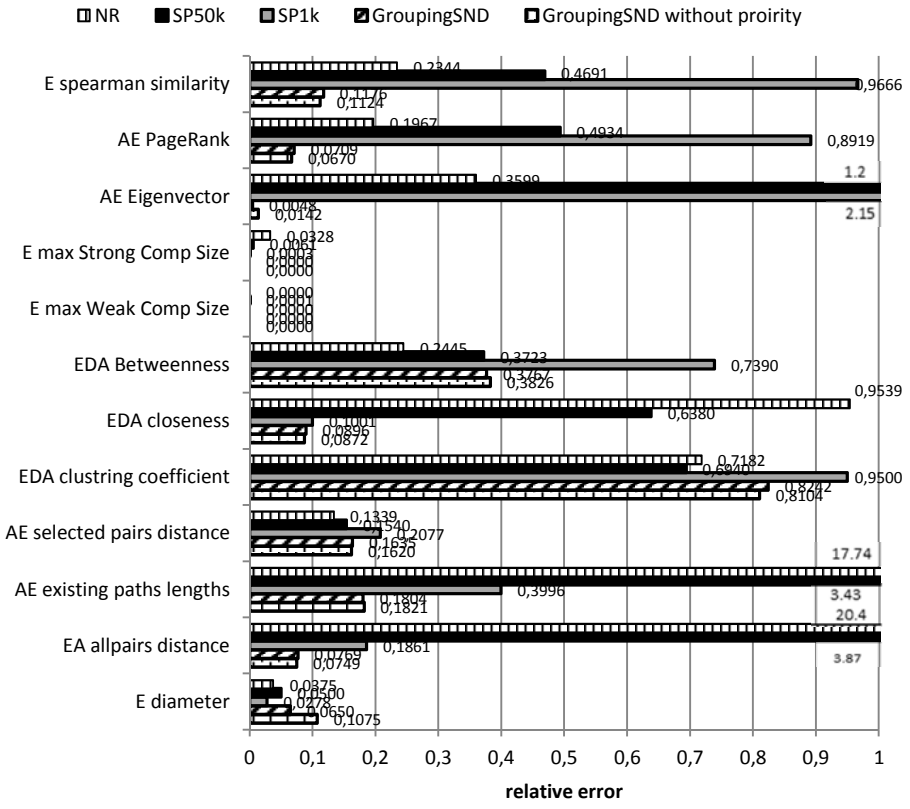


Figure 12: comparison of our method with SP and NR

There is always a trade-off between privacy and utility. Increasing privacy requirements decrease data utility. SP and NR only consider structural data and only protect against link disclosure. The proposed method considers both tabular and structural data and protects against four types of disclosure. Although the proposed method is not absolutely superior [Fig. 12], but the proposed method established a better trade-off between privacy and utility over the results of SP and NR

7 Conclusion and Future Work

We presented a novel greedy algorithm based on a new desirability metric to generate anonymized groups of the ASN technique and the (α,β,γ,δ)-SNP model [Rajaei 2013]. This algorithm tries to preserve structural and tabular data utility while it satisfies all four privacy constraints of (α,β,γ,δ)-SNP using two novel metrics: (1) *privacy metrics* for each privacy requirement (α-presence, β-sensitive-association, γ-degree-association, δ-relationship) measure the desirability of two individuals being located in the same group; (2) *utility metrics* measure the similarity of individuals located in the same group to preserve data utility for topological and aggregate network queries.

These two metrics are denoted as *desirability metric* used in the proposed grouping method to generate groups of minimal size. In addition, methods were introduced based on ASN to measure the aggregate tabular and network queries on the published data. A new utility metric, EDA, was provided to evaluate information loss of topological properties which considers the average absolute difference of all individuals.

Experimental results on three datasets demonstrated that the proposed anonymization technique (ASN) and algorithm preserved data utility at a satisfactory level in all four types of query (aggregate tabular, aggregate network, graph topological, spectrum properties). Since, the proposed method uses directed network data, the only options for comparison of the proposed algorithm for structural properties were SP [Milani Fard 2012] and NR [Milani Fard 2013]. In contrast to the proposed method, these methods only consider structural data and only protect against relationship disclosure in a directed network. Experimental results showed that while the proposed method protects against disclosure of the more private information, it preserves most structural properties better than or similarly to SP and NR. As a result, the proposed method provides a better trade-off between privacy and utility than did SP and NR.

An important area for future study is improvement of data utility using learning methods to find suitable coefficients (F_{QI}, F_S, F_{Din}, F_{Dout} and F_{succ}) based on the original network data distribution. These coefficients effect the calculation of the DM. [Fig. 6(b)] shows that, in some cases, the average probability of disclosure of each privacy requirement was much lower than the required privacy thresholds. In privacy model, it is sufficient to maintain these probabilities near their thresholds. We plan to design a grouping algorithm with few nodes and low differences between disclosure probabilities and privacy thresholds that can generate groups with more similar nodes than the proposed greedy algorithm. In this way, information loss decreases and privacy requirements are protected at the specified thresholds with the same group sizes.

Every social network can be anonymized using the proposed anonymization technique if the thresholds of β, γ, and δ are greater than the density of the related properties in the dataset. In the worst case, all individuals are located in one group and this group satisfies the privacy constraints. Some anonymization methods, especially the greedy algorithm, may not able to find this grouping. For example, in the proposed anonymization algorithm, a node may remain in the last group, but appending it to each generated group violates privacy constraints. In this case two groups should be merged and the node added to the newly-merged group. Future plans include the design of a grouping algorithm based on evolutionary algorithms to provide better grouping.

Another area for future work is development of a grouping algorithm for *ASN* technique that considers different privacy thresholds for different individuals.

References

[Barbaro 2006] Barbaro, M., Zeller, T.: "A face is exposed for AOL searcher"; no. 4417749. New York Times (2006, August 9).

[Bhagat 2009] Bhagat, S., Cormode, G., Krishnamurthy, B., Srivastava, D.: "Class-based graph anaonymization for social network data"; Proceedings of the VLDB Endowment, 2, 1 (2009), 766-777.

[Bonchi] Bonchi, F., Gionis, A., Tassa, T.: "Identity Obfuscation in Graphs Through the Information Theoretic Lens"; Proc. ICDE'11, IEEE, Washington (2011), 924-935.

[Campan 2008]]Campan, A., Truta, T. M.: "A clustering approach for data and structural anonymity in social networks"; Proc. PinKDD'08, ACM, Las Vegas (2008).

[Cha 2007] Cha, S. H.: "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions"; INT. J. OF MATH. MODELS & METHODS IN APP. SCI., 1, 4 (2007), 300-307.

[Cheng 2010] Cheng, J., Fu, A. W., Liu, J.: "K-Isomorphism: Privacy Preserving Network Publication against Structural attacks"; Proc. SIGMOD'10, ACM., USA (2010) 459-470.

[Cormode 2010] Cormode, G., Srivastava, D., Yu, T., Zhang, Q.: "Anonymizing bipartite graph data using safe groupings"; VLDB J., 19, 1 (2010), 115-139.

[Fung 2010] Fung, B. C., Wang, K., Chen, R., Yu, P. S.: "Privacy-preserving data publishing: A survey on recent developments"; ACM Computing Surveys, 42, 4 (2010).

[Hay 2010a] Hay, M.; "Enabling Accurate Analysis of Private Network Data"; Open Access Dissertations. Paper 319, http://scholarworks.umass.edu/open_access_dissertations/319 (2010).

[Hay 2010b] Hay, M., Miklau, G., Jensen, D., Towsely, D., Li, C.: "Resisting Structural Re-identification in Anonymized Social Networks"; VLDB J., 19, 6 (2010), 797-823.

[Hay 2007] Hay, M., Miklau, G., Jensen, D., Weis, P., Srivastava, S.: "Anonymizing social networks"; University of Massachusetts Technical Report Amherst, March (2007).

[Kleinberg 2007] Kleinberg, J. M.; "Challenges in mining social network data: processes, privacy, and paradoxes"; Proc. KDD'07, ACM, USA (2007). 4-5.

[Liu 2008] Liu, K., Terzi, E.: "Towards identity anonymization on graphs".; Proc. SIGMOD'08, ACM Press, New York (2008), 93-106.

[Machanavajjhala 2007] Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: "l-diversity: Privacy beyond k-anonymity"; ACM Trans. Knowl. Discov. Data, 1, 1 (2007).

[Medforth 2011] Medforth, N., Wang, K.: "Privacy Risk In Graph Stream Publishing For Social Network Data"; Proc. ICDM'11, IEEE (2011).

[Milani Fard 2013] Milani Fard, A., Wang, K.: "Neighborhood Randomization for Link Privacy in Social Network Analysis"; The World Wide Web Journal , 16, 5 (2013).

[Milani Fard 2012] Milani Fard, A., Wang, K., Yu, P. S.; "Limiting link disclosure in social network analysis through subgraph-wise perturbation"; Proc. EDBT'12. Berlin, Germany (2012) 109-119.

[Rajaei 2013] Rajaei, M & ,.Haghjoo, M. S.: "An anonymization technique to protect against presence, sensitive-assosiation, degree-assosiation and relationship disclosure in directed social network data publication"; Proc. The International ISC Conference on Information Security and Cryptology .Yazd, Iran. (2013), In Persian.

[Sweeney 2002] Sweeney, L: "Achieving k-anonymity privacy protection using generalization and suppression"; Int. J. Uncert. Fuzz. Knowl,-based Syst, 10 (2002) 571-588.

[Srivastava 2008] Srivastava, J., Ahmad, M. A., Pathak, N., Hsu, D. K.-W.: "Data mining based social network analysis from online behavior"; Proc. SDM'08, (2008).

[Tai 2014] Tai, C. H., Yu, P. S., Yang, D. N., Chen, M. S.; "Structural Diversity for Resisting Community Identification in Published Social Networks". IEEE Trans. Knowl.& Data. Eng., 26, 1 (2014), 235-252 .

[Wang 2010] Wang, H.: "Privacy-Preserving Data Sharing in Cloud Computing"; Journal of conputer Science and technology , 25 ,3 (2010), 401-414.

[Wu 2010a] Wu, L., Ying, X., Wu, X.; "Reconstruction from randomized graph via low rank approximation"; Proc. SDM'10, Columbus (2010).

[Wu 2010b] Wu, W., Xiao, Y., Wang, W., He, Z., Wang, Z.; "k-symmetry model for identity anonymization in social networks"; Proc. EDBT'10. USA: ACM (2010), 111-122.

[Wu 2009] Wu, X., Ying, X., Liu, K., Chen, L.: "A Survey of Algorithms for Privacy-Preservation of Graphs and Social Networks". In C. C. Aggarwal , & H. Wang (Eds.), Invited book chapter. Managing and Mining Graph Data. Kluwer Academic Publishers (2009).

[Xiao 2006] Xiao, X., Tao, Y.: "Anatomy: Simple and effective privacy preservation"; Proc. VLDB'06. Seoul, Korea, (2006), 139-150.

[Ying 2009] Ying, X., Wu, X.: "On link privacy in randomizing social networks"; Lecture Notes in Computer Science, 5476 (2009), 28-39

[Ying 2008] Ying, X., Wu, X.: "Randomizing social networks: a spectrum preserving approach". Proc. SDM'08 SIAM (2008), 739-750.

[Yuan 2013] Yuan, M., Chen, L., Yu, P. S., Yu, T.: "Protecting Sensitive Labels in Social Network Data Anonymization"; IEEE TRANS. KDE, 25, 3 (2013), 633-647.

[Zhang 2007] Zhang, Q., Koudas, N., Srivastava, D.,Yu, T.: "Aggregate query answering on anonymized tables"; Proc. ICDE07, IEEE, Istanbul, Turkey (2007), 116-125.

[Zhou 2011] Zhou, B., Pei, J.: "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks"; An Int. J. KAIS , 28, 1 (2011), 47-77.

[Zou 2009] Zou, L., Chen, L., Ozsu, M. T.; "K-automorphism: A general framework for privacy preserving network publication"; Proc. VLDB09. ACM (2009), 946-957.

---

[1] http://jung.sourceforge.net/
[2] http://snap.stanford.edu/data/wiki-Vote.html
[3] http://deim.urv.cat/~aarenas/data/xarxes/email.zip
[4] http://pajek.imfm.si
[5] http://www.ipums.org
[6] $p(k) \approx k^{-\gamma}$, where p(k) denotes the fraction of nodes with degree k, and γ is constant value typically in the range $2<\gamma<3$